60407600

CONTROL DATA CORPORATION

KRONOS® 2.1 TIME-SHARING USER'S REFERENCE MANUAL

CONTROL DATA®
CYBER 70 SERIES
MODELS 72, 73, 74
6000 SERIES
COMPUTER SYSTEMS

COMMAND INDEX

Command	Page
APL	4-9
APPEND	5-6
ASCII	4-2
ATTACH	5-6
AUTO	4-3
BASIC	4-6
BATCH	4-7
BINARY	4-10
BYE	4-10
CATLIST	5-12
CHANGE	5-7
CHARGE	4-11
CLEAR	4-11
CONVERT	4-11
CSET	4-4
DAYFILE	4-11
DEFINE	5-8
EDIT	4-11
ENQUIRE	4-14
EXECUTE	4-7
FTNTS (FORTRAN)	4-8
FULL	4-4
GET	5-9
GOODBYE	4-11
HALF	4-4
HELLO	4-11
HELP	4-12
I key	4-12
LENGTH	4-12
LIBRARY	4-13
LIMITS	4-13
LIST	4-14
LNH	4-14
LOGIN	4-14
NEW	4-14
ND	4-15
NORMAL	4-5
NOSORT	4-15
NULL	4-9
OLD	4-15
PACK	4-15
PACKNAM	5-10
PARITY	4-5
PASSWOR	4-16
PERMIT	5-11
PRIMARY	4-16
Procedure File Call (-)	4-22
PURGE	5-11
RECOVER	4-16
RENAME	4-16
REPLACE	5-11
RESEQ	4-16
RETURN	4-17
REWIND	4-17
RNH	4-19
ROUT	4-5
RUN	4-17
S key	4-19
SAVE	5-12
SETTL	4-19
SORT	4-19
STOP	4-21
SUBMIT	4-21
TAPE	4-5
TERM	4-5
TEXT	4-21

60407600

CONTROL DATA CORPORATION

KRONOS® 2.1 TIME-SHARING USER'S REFERENCE MANUAL

CONTROL DATA® CYBER 70 SERIES MODELS 72, 73, 74 6000 SERIES COMPUTER SYSTEMS

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released.
(7-1-73)	
B	Manual revised. This edition amplifies and updates the descriptions of system (section 4) and
(5-1-74)	permanent file commands (section 5). Typographical errors have been corrected throughout.
C	Manual revised. This edition outlines corrections to log-in procedure, adds information on
(12-9-74)	additional time-sharing commands, and gives details on the FTNTS subsystem which replaces
	the FORTRAN subsystem.
D	Add KRONOS 2.1.2 features (PSRs 404/401). This edition obsoletes all previous editions.
(6-17-75)	

Publication No.
60407600

© 1973, 1974, 1975
by Control Data Corporation

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

or use Comment Sheet in the back of this manual.

PREFACE

The KRONOS® Time-Sharing System was developed by Control Data Corporation to provide remote interactive job processing for CONTROL DATA® CYBER 70 Series Model 72, 73, and 74 Computer Systems and for CONTROL DATA® 6000 Series Computer Systems. This interactive job processing capability is provided in addition to the local and remote batch processing capabilities available under KRONOS.

This manual describes communication between a remote time-sharing terminal user and the KRONOS Time-Sharing System. It contains a description of the subsystems available to the user and the procedures necessary to use the capabilities of these subsystems. However, this manual does not contain programming information for these subsystems. Sections 4 and 5 of this manual may be particularly useful to the inexperienced user as well as the more experienced user. These sections describe the command set available to all time-sharing users. Included are terminal control, subsystem selection, file creation, and file maintenance commands. The more experienced user may be interested in the ability to enter control card images, either one at a time, or as a batch job image, from the terminal. These features are described in sections 8 and 9.

The reader should note that this product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

The following manuals contain additional information of interest to the user of the KRONOS Time-Sharing System.

Control Data Publication	Pub. No.
KRONOS 2.1 Reference Manual	60407000
KRONOS Instant Manual	60407200
KRONOS Terminal User's Instant Manual	60407800
KRONOS Time-Sharing FORTRAN Reference Manual	60408600
APL*CYBER Reference Manual	19980400
KRONOS EXPORT/IMPORT Reference Manual	59150500
KRONOS Text Editor Reference Manual	60408200
KRONOS Transaction Subsystem Reference Manual (TRANEX)	60407900
KRONOS MODIFY Reference Manual	60281700
KRONOS MODIFY Instant Manual	60283000
KRONOS General Information	60407100
KRONOS Record Manager Reference Manual	60385400

The following manuals describe the products common to KRONOS and the CONTROL DATA® SCOPE operating system.

Control Data Publication	Pub. No.
COMPASS 3.1 Reference Manual	60360900
FORTRAN 2.3 Reference Manual	60174900
FORTRAN Extended 4.4 Reference Manual	60305600
COBOL 4.4 Reference Manual	60384100
ALGOL 3.0 Reference Manual	60329000
Simula 1.1 Reference Manual	60234800
SIMSCRIPT 3.0 Reference Manual	60178300
Sort/Merge 4.2 Reference Manual	60343900
PERT/Time 1.2 Reference Manual	60133600
Update Reference Manual	60342500
BASIC 2.1 Reference Manual	19980300

CONTENTS

SECTION 1	INTRODUCTION	1-1
	Terminals	1-1
	Subsystems	1-2
	Conventions	1-2
SECTION 2	INTRODUCTION TO KRONOS PROCESSING	2-1
	Log-in Procedure	2-1
	Program Entry and Execution	2-6
	Log-off Procedure	2-7
SECTION 3	FILE USAGE	3-1
	File Structure	3-1
	Working Files	3-2
	File Sorting	3-3
	Working File Control	3-6
SECTION 4	SYSTEM COMMANDS	4-1
	Command Processing	4-1
	Terminal Control Commands	4-2
	Subsystem Selection Commands	4-7
	Time-Sharing Job Commands	4-10
	LIMITS Command	4-25
SECTION 5	PERMANENT FILE COMMANDS	5-1
	Permanent File Command Format	5-2
	Permanent File Processing Commands	5-6
	CATLIST Command	5-12
	Alternate User Access	5-18
	LIBRARY Command	5-22
SECTION 6	TIME-SHARING JOB PROCESSING	6-1
	Input/Output Conventions	6-1
	Terminal Job Processing	6-4
	Terminal Log-In	6-4
	Subsystem Selection	6-4
	Program Entry and Execution	6-7
	Terminal Log-Off	6-8
SECTION 7	ADDITIONAL SYSTEM FEATURES	7-1
	Program Editing	7-1
	Resequencing Line Numbers	7-2
	Job Suspension	7-3
	Recovery	7-5

SECTION 8	REMOTE JOB ENTRY Reformatting the Submit File Error Processing	8-1 8-2 8-5
SECTION 9	BATCH SUBSYSTEM	9-1
GLOSSARY		Glossary-1
APPENDIX A	TERMINAL ERROR MESSAGES	A-1
APPENDIX B	DIAL-IN PROCEDURE AND CHARACTER SET	B-1
APPENDIX C	PROGRAM CONTROL OF TERMINAL ACTIVITY	C-1
APPENDIX D	PAPER TAPE OPERATIONS	D-1
APPENDIX E	MASS STORAGE DEVICE STATISTICS	E-1
APPENDIX F	TEXT EDITOR COMMAND INDEX	F-1
APPENDIX G	RESERVED FILE NAMES	G-1
APPENDIX H	SAMPLE SESSION AT THE TERMINAL	H-1

INTRODUCTION 1

The KRONOS Time-Sharing System provides a user with general purpose or specialized time-sharing capabilities. It can be used from any location simply by dialing the computer's telephone number.

Since KRONOS is capable of serving a large number of users simultaneously, the cost of using the system is only a fraction of that of conventional data processing systems. This is combined with the speed and convenience of a conversational system.

Although a large number of users may be accessing KRONOS at the same time, the sophisticated software that controls the system assures the complete separation of each activity. Thus, each user may conceptually imagine that he is the only user of the system.

To further enhance the convenience of programming, the KRONOS system maintains control of all information (files) created by its users. Prior to creating a file, it is named and may subsequently be requested by that name at any time.

The KRONOS system is inherently secure. It assumes that any file created belongs exclusively to the person creating it and will deny any other user's request to access it. However, a user can also specify that a file be made available to other users.

In order to use the system, it is necessary to know its telephone number and be able to supply identification consisting of a user number and optionally, a password, charge number, and project number. After dialing the appropriate telephone number and making connection with the computer, the system requests the user indentification. In some cases, the terminal being used may also require identification. If the identification is legitimate, access to the KRONOS system occurs immediately.

In summary, the KRONOS system is a powerful, sophisticated time-sharing system which can satisfy a wide spectrum of computational needs of its users. Its conversational capabilities permit its users to debug their programs much faster than with other data processing systems. Its peripheral mass storage and advanced file maintenance techniques permit vast amounts of information to be stored at high speeds. The fact that the system can be used from terminals in remote locations creates an environment in which many users can be performing work in the privacy of their own offices, without the inconvenience of traveling to the computer site.

TERMINALS

All communication with the computer can take place via a remote terminal. Programs or data are sent to the computer according to specific rules or commands. The computer responds by sending its answers on the terminal. If the user attempts to process an incorrect command, the computer rejects the command and prints the reason for its rejection.

Two types of terminals may be used to communicate with KRONOS.

- ASCII code compatible terminals (such as Teletype® models 33, 35, 37, and 38, Memorex 1240, etc.)
- Correspondence code terminals (such as Novar 5-41, Datel 30, IBM 2741, etc.)

If a correspondence code terminal is to be used, KRONOS may require that the communications system equipment be modified to accommodate this type of terminal. In addition, modification may also be required if transmission speeds greater than ten characters per second are required.

Most time-sharing operations can be performed using either type of terminal. Differences in operation are described, when applicable, throughout this manual. A summary of these differences is described in Table B-1 of appendix B.

SUBSYSTEMS

After identifying himself to the system, the user can specify a subsystem to be used. The available subsystems are:

- BASIC — An elementary programming language used for scientific applications. This easily learned language is quickly adaptable to the needs of engineers, businessmen, educators, and other professionals. It requires neither computer background nor training in higher mathematics.
- BATCH — Provides the user with a batch control card capability from the terminal. It enables a user to type control card images at his terminal that would normally have to be entered from a card reader at the central site or from a remote batch environment.
- EXECUTE — Allows a user to execute a previously compiled program. Since a majority of the time spent on a job is for compilation, this subsystem provides an efficient means to run a frequently used program without having to recompile it each time.
- FTNTS — Translates mathematical language into machine language to solve mathematical and engineering problems. The FORTRAN time-sharing subsystem provides a convenient language for expressing mathematical and scientific problems and a convenient method for data input and output.
- NULL — Allows the user to perform file manipulations and other time-sharing operations without subsystem association.

CONVENTIONS

Throughout this manual, the following conventions are used.

- (CR) denotes the RETURN key on the terminal keyboard. Each line of time-sharing terminal input must be followed by a (CR). The system responds by performing a carriage return and line feed operation (positions carriage to first character position on the next line).
- The word system refers to the KRONOS Operating System which provides the interface between KRONOS and the user's program.
- The vertical spacing in examples does not necessarily coincide with the spacing that will appear on the user's terminal.

It is recommended that a new user refer to the glossary to become familiar with terms used in this manual and other KRONOS manuals.

INTRODUCTION TO KRONOS PROCESSING 2

This section is intended primarily for users unfamiliar with the KRONOS Time-Sharing System. It is designed as a guide to be used for the first session at a time-sharing terminal. Included are instructions necessary to gain access to the system (log-in), enter and run a simple program in the BASIC language, and finally, leave the system (log-off). You cannot harm the system by making incorrect entries or other mistakes at your terminal. The error will merely be detected by the system which in turn issues a routine error message (probably ILLEGAL COMMAND) back to your terminal. Generally, you are allowed as many chances to make a correct entry as are necessary. However, if you are unsuccessful at logging into the system four times in succession, your terminal is automatically disconnected. If this happens, merely recheck your assigned user number and password (if one is assigned), and start over.

LOG-IN PROCEDURE

1. Check to ensure that the following switches on the terminal are set to the correct position. It is possible that a terminal does not contain one or more of the switches listed. If the terminal has been determined to be acceptable for KRONOS time-sharing operations, it use should correspond to that described in this manual.

   - Full-duplex/half-duplex (FDX/HDX) switch - Although this switch can be set to either position, it is normally set to the half-duplex position. If set to full-duplex, characters entered from the terminal are not printed.
   - Even parity/odd parity switch - If an ASCII code terminal is being used, this switch must be set to even parity. Odd parity must be selected for correspondence code terminals.
   - Baud rate switch - This switch determines the speed at which the terminal transmits and receives information. There is only one correct position when the terminal is connected to KRONOS. However, this is a system option and is dependent upon the system equipment being used. If you do not know the correct setting, consult your service vendor.

2. Dial the number of the KRONOS computer site. This procedure is detailed in appendix B. When this connecting sequence is complete, KRONOS will either print out the header information, or if terminal identification is required, it will do nothing. Correspondence code terminals must always be identified. Specific ASCII code terminals may also require identification. If terminal identification is not required, go to step 4. If it is required, go to step 3.

3. If terminal identification is required, perform one of the following:

- Correspondence code terminals with standard print

  Press the ATTN key

- Correspondence code terminals with standard print, 63-character set†

  Type D and press the ATTN key

- Correspondence code terminals with APL print

  Type A and press the ATTN key

- Correspondence code terminals with APL print, 63-character set†

  Type alpha and press the ATTN key

- ASCII code terminals with standard print

  Type (CR)

- ASCII code terminals with standard print, 63-character set†

  Type D

- Memorex 1240 (ASCII code) terminals with APL print

  Type M

- Block transmission terminal

  Type B

- Memorex 1240 (ASCII code) terminal with APL print, 63-character set†

  Type P

- ASCII code with standard print, block edit mode

  Type B

- ASCII code terminals with standard print, block edit mode, 63-character set†

  Type b

† 63-character set is provided for compatability with previous systems.

4. KRONOS initiates the log-in sequence by printing the header identification either automatically (after step 2) or in response to terminal identification (after step 3). The format of this header is:

```
yy/mm/dd.     hh.mm.ss.
KRONOS TIME-SHARING SYSTEM - VER. 2.1.2
```

where:

yy/mm/dd	Current date (year/month/day.)
hh.mm.ss.	Current time (hours.minutes.seconds.)

5. If the system contains more than permanent file family, the system requests the name of the family under which the user's permanent files exist or will exist by responding with:

```
FAMILY:
```

Enter the family name on the same line and press carriage return.

If the user's family is the same as the system default family, type (CR) only.

If no family identification is required, omit this step.

6. The system requests a user number by printing:

```
USER NUMBER:
```

Type user number on the same line and press carriage return.

7. The system then requests entry of a password.

```
PASSWORD
XXXXXXXX
```

The cursor is positioned at the beginning of the blacked-out area. Type in the password and enter carriage return. If no password is in effect, press carriage return only.

The password can be entered immediately after the user number in step 6. In such a case, the system skips step 7. The family name, user number, and password can also be entered on one line.

```
FAMILY: SYS100,USERNUM,PASSWRD (CR)   : is acceptable
and equivalent to:
FAMILY: SYS100 (CR)
USER NUMBER: USERNUM (CR)
PASSWORD:
XXXXXXXX (CR)
```

8. The system then prints either:

```
TERMINAL: nnn,iii                 TERMINAL: nnn,iii
RECOVER/CHARGE:        or         RECOVER/SYSTEM:
```

nnn Terminal number that the system uses as part of the unique identifier for each active user

iii Identifier that describes the type of terminal as follows:

TTY	ASCII code terminal with standard print
TTYD	As TTY but with 63-character set
COR	Correspondence code terminal with standard print
CORD	As COR but with 63-character set
CORAPL	Correspondence code terminal with APL print
CORAPLD	As CORAPL but with 63-character set
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print
MEMAPLD	As MEMAPL but with 63-character set
BLKEDT	Block transmission terminal with full display screen capability (available only on certain terminals such as Hazeltine 2000)
BLKEDTD	As BLKEDT but with 63-character set

The TTYD, MEMAPLD, CORAPLD, CORD and BLKEDTD translation tables are provided for compatability with previous systems.

If, instead of RECOVER/SYSTEM or RECOVER/CHARGE, the message SYSTEM CLOSED, SORRY is received, the user should stop. This message means that the number of users logged in has exceeded the maximum number allowed. This number can be 0 and is used when the operator wishes to close the whole system to terminal users. Users logged onto the system are not affected. The user must wait and try again later.

9. Perform step a or b depending upon which system response was printed in step 8.

   a. If the RECOVER/SYSTEM: response was printed, enter the subsystem to be used on the same line (BASIC for this session). Recovery is discussed in section 7.

For example:

BASIC (CR)

The system responds by printing:

OLD, NEW, OR LIB FILE:

A subsystem may be automatically selected by the system at log-in if this has been specified in the system file. This validation file determines the internal system controls associated with each valid user number. Refer to the discussion of the LIMITS command at the end of section 4 for further information.

b. If the RECOVER/CHARGE: response was printed, type CHARGE, followed by your assigned charge number and project number on the same line.

For example:

CHARGE, chargeno, projectno (CR)

The system responds by printing:

READY.

Enter the subsystem to be used (BASIC for this session).

For example:

BASIC (CR)

The system responds by printing:

OLD, NEW, OR LIB FILE:

10. Enter the appropriate file status on the same line (NEW for this session).

For example:

NEW (CR)

The system responds by printing:

FILE NAME:

11. Enter the one-to seven-character file name on the same line (PRIME for this session).

For example:

PRIME (CR)

The system responds by printing:

READY.

At this point, the output at the terminal has the following form.

```
yy/mm/dd.  hh.mm.ss
KRONOS TIME-SHARING SYSTEM - VER. 2.1.2
FAMILY:  SYS101
USER NUMBER:  usernum
PASSWORD
XXXXXXXXX                          Password, if present, is not
TERMINAL:  nnn, iii                visible through blackout.
RECOVER/SYSTEM:  BASIC
OLD, NEW, OR LIB FILE:  NEW
FILENAME:  PRIME
READY.
```

Now, if the user wishes, he can vary the source and format of information sent to or received from the system by entering any of the terminal control commands (section 4). All of these commands, except AUTO and TAPE, may be automatically selected by the system at log-in, depending on the validation parameters that have been established for the user's particular site.

PROGRAM ENTRY AND EXECUTION

1. Now enter the BASIC program PRIME. Each program statement is followed by a carriage return.

```
10 PRINT "THIS IS A LISTING OF TEN PRIME NUMBERS"
11 PRINT
20 PRINT "WHICH PRIME          THE PRIME NUMBER"
30 LET N=0
40 LET V=0
45 FOR A=1 TO 10
50 LET V=V+1
60 LET Y=INT(V/2+.5)
70 FOR X=2 TO Y
80 LET Z=V/X
90 LET W=INT(Z)
100 IF Z=W THEN 50
110 NEXT X
120 LET N=N+1
140 PRINT"     ";N,"              ";V
160 NEXT A
170 END
```

2. The program is now ready to be executed. Type.

   RUN (CR)

   The system returns the following output.

```
yy/mm/dd.    hh.mm.ss.
PROGRAM    PRIME

THIS IS A LISTING OF TEN PRIME NUMBERS

WHICH PRIME          THE PRIME NUMBER
     1                      1
     2                      2
     3                      3
     4                      5
     5                      7
     6                     11
     7                     13
     8                     17
     9                     19
    10                     23
```

3. This program prints the first ten 10 prime numbers. After the program has completed execution, the system replies:

```
SRU   s.sss UNTS
RUN COMPLETE.
```

Where:

s.sss	Amount of system resource units (SRU) required to compile and execute the program. An SRU is a measure of CPU time, memory usage, and I/O time. If the user logged in with a charge/project number, the total SRUs accumulated for that charge/project number are given. If the user did not log-in with a charge/project number, the number of SRUs since log-in are given.

LOG-OFF PROCEDURE

To leave the system, type:

BYE (CR)

This command causes the following information to be printed, and then logs you off the system by disconnecting the terminal.

```
USER123   LOG   OFF.   hh.mm.ss
USER123   SRU          s.sss UNTS.
```

s.sss	Total number of SRUs used under this charge/project number

TIME OUT

The system logs-off an inactive terminal if no entry is made within a 10-minute period.

FILE USAGE 3

With the exception of some central memory tables, all data in the KRONOS system is considered either a file or a part of a file. This section describes the mass storage file organization under KRONOS and explains the fundamentals of file processing. Refer to the KRONOS Reference Manual for detailed information concerning the KRONOS file structure.

FILE STRUCTURE

A file is the largest collection of information addressable by name (one to seven characters), and begins with the beginning-of-information (BOI), before which no data exists. A file consists of one or more logical records of information. A logical record is a group of related words or characters, of fixed or variable length, which is independent of its physical environment. Thus, a logical record on punched cards would contain the same information as its counterpart on mass storage.

The physical record unit (PRU) is the smallest division of data on a device. A PRU on all mass storage devices in the KRONOS system consists of 640 six-bit characters. The logical record is composed of one or more PRUs.

The end of a logical record is the end-of-record (EOR). The end of a file is the end-of-information (EOI). The system automatically writes an EOI as the last physical item of information on all files. However, the user may have the additional capability to write an end-of-file mark (EOF) to signify the end of a file. This capability exists in the FORTRAN (FTNTS) subsystem and in other languages available under the BATCH subsystem (COMPASS, COBOL, etc.). Therefore, due to the EOI concept, a file may actually be a multifile file. For example:

(BOI)...data...(EOR)...data...(EOR)(EOF)...data...(EOR)(EOF)(EOI)

A typical file is illustrated by the following example.

> One line of an invoice may form an item, a complete invoice may form a record, a set of such records may form a file, and the collection of invoice files may form a multifile file.

To have a file positioned at the BOI means that a subsequent read operation will read the first logical record. A write operation writes data on the first PRU and terminates with an EOR and EOI following the logical record. A file positioned at EOR has the following record, EOF, or EOI read on a subsequent read operation. A subsequent write operation on the file adds data to a new physical record following the current EOR or EOF. A file positioned at EOI is positioned at the last physical record of the file. Any subsequent read operation causes the EOI to be read. No data is transmitted. A write operation writes over the existing EOI and adds a new logical record to the file followed by an EOR and EOI.

The following examples illustrate possible positions within a file.

File 1 ...data...(EOR)(EOF)(EOI)

[1] [2] [3] [4]

File 2 ...data...(EOR)...data...(EOR)(EOF)...

[1] [2] [3]

File 3 ...data...(EOR)(EOI)

[1] [2] [4]

[1] File positioned at BOI

[2] File positioned at EOR

[3] File positioned at EOF

[4] File positioned at EOI

WORKING FILES

Generally, a working file is either a new file, created by the user, or a copy of a file that already exists in the system (indirect access permanent file†). All working files are temporary in nature and can exist no longer than the user is logged into the system. Working files may be created, accessed, and released at the discretion of the user while he is logged into the system but they are automatically released when he logs off the system.

Working files may also be referred to as local files. The parameter lfn on most time-sharing job commands and permanent file commands (described in sections 4 and 5) signifies a local file name. Therefore, throughout this manual, the terms are synonymous.

One type of working file, the primary file, has special significance in certain time-sharing commands. A primary file is obtained with the OLD or LIBRARY command which retrieves a copy of an indirect access permanent file. In addition, a primary file is created with the NEW command. There is only one primary file active or available to the user at any given time. For example, the command

```
OLD,LFN1
```

retrieves a copy of indirect access permanent file LFN1 from permanent file storage for use as the primary file. If the command

```
OLD,LFN2
```

is entered, a copy of indirect access permanent file LFN2 is retrieved for use as the primary file, and the current primary file (LFN1) is released. To access LFN1 again, the user must enter the OLD command or enter:

```
GET,LFN1
```

If this command is used, a copy of permanent file LFN1 is retrieved for use as a working file and the current primary file (LFN2) remains the same.

---

†Refer to section 5 for complete information concerning permanent files.

Most operations performed on files by time-sharing terminal commands are performed using the primary file, unless another working file is specified in a command parameter. For example, the command

SAVE

is used to retain the primary file in the permanent file system. However, the command

SAVE,ABC

is used to retain file ABC in the permanent file system. File ABC could be either the primary file or another working file.

There is a close association between the primary file and the subsystem in use when the primary file is saved. Unless the NULL subsystem is active when the primary file is saved, an internal indicator called the subsystem flag is set to indicate which subsystem is being used. That subsystem becomes associated with the permanent file and is automatically selected each time the file is retrieved using the OLD command. For example, if a user currently operating under the BASIC subsystem issues the command

SAVE,ABC

where ABC is the current primary file name, the BASIC subsystem flag will be associated with the resulting permanent file ABC. If at a later date the user is operating under another subsystem and enters the command

OLD,ABC

the current primary file is released, the BASIC subsystem is selected automatically, and file ABC becomes the new primary file. To save the primary file without a subsystem association, the user must enter the NULL subsystem before issuing the SAVE command. Retrieving a file saved while under the NULL subsystem does not affect the subsystem currently being used. Furthermore, the subsystem flag is set only when saving the primary file since this is the only working file that has a subsystem associated with it. If the permanent file named in the command to select a new primary file (OLD or LIBRARY) does not have a subsystem associated with it, the subsystem currently being used remains in effect.

Following any operation on a primary file, the file is automatically rewound (positioned at BOI). However, the user is responsible for the position of all other working files because they are not automatically rewound following each operation. The REWIND command from a time-sharing terminal, the REWIND statement in FTNTS, or the RESTORE statement in BASIC can be used to position a file to the BOI.

Information (program statements, text, or data) entered from the terminal keyboard is normally stored in the primary file. However, the user can also enter source code into any working file, including the primary file, or create new working files using the Text Editor. Refer to the Text Editor (EDIT) Reference Manual for additional information.

FILE SORTING

Each line of information entered by the user must begin with a line number unless it is entered in text mode. A space must appear between the number and the information which follows. Line numbers for FORTRAN Extended or BASIC programs cannot be more that five digits. All other information can have line numbers up to ten digits. Leading zeros on line numbers are permissible but are not required (00120 and 120 are interchangeable). Line numbers with leading zeros can be intermixed with line numbers containing no leading zeros. Line numbers can be entered manually or may be generated automatically by the system. This is accomplished by entering the AUTO command (refer to description in section 4). Note that the line numbers generated by the AUTO command consist of only five digits.

---

† Refer to the TEXT command in section 4 for information concerning text mode.

When the user is not in text mode, entering numbered lines of information sets an internal indicator called the sort flag. When the sort flag is set, the primary file is sorted automatically whenever the LIST, LNH, RUN, REPLACE, RNH, LENGTH, EDIT, SAVE, or SUBMIT command is entered. Sorting of the file places the statement in order according to the first five digits of the line number. If it is desired to sort the file on more than the first five digits, the number of digits can be specified using the SORT, lfn command (NC parameter). Refer to description of this command in section 4.

When entering information from the terminal, the user enters the line number and the statement. After the carriage return (CR) key is pressed, the text is entered into an area in core. Periodically, when this area (16 decimal central memory words) is filled, the text is written to the user's primary file on mass storage as a new logical record. When a sort is performed, these records are rewritten as one logical record. Thus, the entire input is stored as one logical record.

If the message

FILE TOO LONG TO SORT.

is returned to the user, the file is too long to be sorted automatically by the system. The user must then enter the SORT, lfn command before proceeding.

However, if the message

FILE NOT SORTED.

is returned to the user, the system has attempted to sort the file but discovered that one or more lines begin with a nonnumeric character. Thus, a sort was not performed and the sort flag was turned off. The user may enter the SORT, lfn command for more informative diagnostics.

If the user wishes to inhibit automatic sorting of the primary file, he may enter the NOSORT command. This command clears the sort flag and should be entered only when the user desires that the primary file not be sorted and packed into one logical record. The sort flag is reset when KRONOS receives the first numbered line of source code following the NOSORT command.

The following example illustrates the effect of the NOSORT command. The user selects the proper subsystem and obtains a copy of permanent file A with the command sequence:

FTNTS, OLD, A

If file A had a subsystem flag other than FTNTS or NULL associated with it, that subsystem would be selected automatically even though the user specified the FTNTS subsystem. To ensure that the correct subsystem and file have been selected, the user enters the command:

ENQUIRE

The system prints the following status report:

```
TERMINAL: 42,     TTY  <----  Identifies the terminal being used
SYSTEM - FTNTS  <-------------  Proper subsystem was selected
FILE NAME: A    <-------------  File A is the new primary file
STATUS - IDLE
```

The user then requests a listing of the file with the command:

LIST

The system prints the following copy of the file.

```
 75/01/10.      16.07.14.
PROGRAM      A

00100 PROGRAM TEST (OUTPUT)
00110 DO 1  J=1, 10
00120 PRINT 2, J
00130 2 FORMAT (2X, I2)
00140 1 CONTINUE
00150 END
READY.
```

The user then enters the following changes:

```
00120 K=J*J
00130 PRINT 2, K
00140 2 FORMAT (2X, I2)
00150 1 CONTINUE
00160 END
```

and enters the command:

```
NOSORT
```

The system responds:

```
READY.
```

The user then calls for a listing with logical record indications by entering:

```
LIST, R
```

The system lists the unsorted file indicating the end of each logical record.

```
 75/01/10.      16.12.23.
PROGRAM      A

00100 PROGRAM TEST (OUTPUT)
00110 DO 1  J=1, 10
00120 PRINT 2, J
00130 2 FORMAT (2X, I2)
00140 1 CONTINUE
00150 END
--EOR--
00120 K=J*J
00130 PRINT 2, K
00140 2 FORMAT (2X, I2)
00150 1 CONTINUE
00160 END
--EOR--
READY.
```

If the user had requested a listing without entering the NOSORT command, the following output would have been provided.

```
 75/01/10.      16.12.23.
PROGRAM      A

00100 PROGRAM TEST (OUTPUT)
00110 DO 1  J=1, 10
00120 K=J*J
00130 PRINT 2, K
00140 2 FORMAT (2X, I2)
00150 1 CONTINUE
00160 END
--EOR--
```

If the user wishes to inhibit file sorting, but desires that the file be packed into one logical record, he may enter the following command.

```
PACK
```

If the user then wishes to perform text editing on the packed file, he can call the Text Editor with the EDIT command (refer to the description of the EDIT command in section 4).

WORKING FILE CONTROL

As mentioned previously, the OLD, NEW and LIBRARY commands change the status of files by exchanging the current primary file for another. When the OLD command is entered, the primary file is exchange for a copy of an indirect access permanent file from the permanent file system. When the NEW command is entered, a new primary file is created which the user fills from the terminal. The LIBRARY command replaces the current primary file with a copy of an indirect access permanent file saved under the special user number LIBRARY.

It is important to note that normally, when the current primary file is released (by entry of the OLD, NEW, or LIBRARY command), all other working files are also released when the next command is entered, unless the ND parameter (NO drop) is specified with the NEW or LIBRARY commands.

The following is a list of sample commands and their effect on files. The example assumes that there is no subsystem association with any file retrieved using the OLD command. Therefore, the subsystem to be used must be specified, if other than the current subsystem.

	Files Present at Completion of Command	
Commands	Primary	Other Working Files
BASIC, OLD, A	A	
GET, B	A	B
GET, C	A	B, C
GET, D	A	B, C, D
FTNTS, OLD, A1 (working files are dropped)	A1	
RUN (program A1 creates TAPE1)	A1	
.		
.		
RUN COMPLETE. (system response)	A1	TAPE1
.		
.		
GET, B1	A1	TAPE1, B1
GET, C1	A1	TAPE1, B1, C1
SAVE, TAPE1	A1	TAPE1, B1, C1
FTNTS, OLD, A2/ND (working files are saved by using ND parameter)	A2	TAPE1, B1, C1
RUN (program A2 creates TAPE2)	A2	TAPE1, B1, C1
.		
.		
.		
RUN COMPLETE. (system response)	A2	TAPE1, TAPE2, B1, C1

Other commands that change the status of working files are:

CLEAR	Releases all working files. Note that only the contents of the current primary file is released; the primary file name is retained. If no primary file is currently defined, a new primary file name is requested.
LOGIN	Releases all working files
HELLO	Releases all working files
GOODBYE	Releases all working files
BYE	Releases all working files

Refer to section 4 for a complete description of these commands.

SYSTEM COMMANDS 4

After the user has successfully logged into the system, he can enter the commands necessary to process his job. There are four general categories of commands available to the user.

- Terminal control commands
- Subsystem selection commands
- Job processing commands
- Permanent file commands

With the exception of permanent file commands, each command category is described in this section. Permanent file commands are described in section 5.

COMMAND PROCESSING

The minimum number of characters required to specify a command depends upon the specific command and the subsystem currently being used. A minimum of three characters are checked when a command is entered in all subsystems except the BATCH subsystem. In the BATCH subsystem, commands cannot be abbreviated. If more than the minimum number of characters are entered, the system checks the number given, up to a maximum of seven characters.

For example, if a user is operating under the BASIC subsystem and wishes to change to FTNTS, the following would be legal and illegal commands for entering FTNTS.

Legal Forms	Illegal Forms	
FTN	F	
FTNT	FT	
FTNTS	FTNx	(where x is any alphanumeric character, except T)

However, if the user is in the BATCH subsystem and wishes to change to any other subsystem, the name of that subsystem must be entered without abbreviation.

In general, KRONOS processes each command by checking from one to seven of the characters entered. If the user does not enter a sufficient number of characters to make the command unique, the system responds

COMMAND NOT UNIQUE.

and the user must then reenter the command using a sufficient number of characters to make it unique. This only occurs with commands in which several characters are identical, such as HELP and HELLO. In this case, HEL would result in the command being declared not unique.

If the command entered is not a valid command or has been misspelled, the message

ILLEGAL COMMAND.

is returned. When the user discovers an error in the command or other input before the carriage return (CR) key is pressed, he may either delete the entire line or backspace to the point of the error, correct it, and continue on the same line.

To delete a line, simply press the ESC or ESCAPE key (ASCII code terminals) or ATTN key (correspondence code terminals). The system responds

DEL

and then positions the carriage to the first character position of the next line.

To backspace and correct a line, count the number of characters (including spaces) past the position of the error and press the backspace key an equal number of times. Then correct the error and continue on the same line. The backspace key on ASCII code terminals is the underline (_) key.† On correspondence code terminals, the BACK SPACE key is used.

If the user attempts to perform an operation for which he is not validated or exceeds his validation limits, the system responds:

ILLEGAL USER ACCESS.

The system maintains validation controls for every user number. Refer to the LIMITS command at the end of this section for a complete description.

Alphabetic characters entered in lowercase are automatically translated to uppercase unless the ASCII command is in effect. The ASCII command inhibits translation from lowercase to uppercase (refer to the description of the ASCII command). Note that all commands may be entered in uppercase or lowercase in either ASCII or non-ASCII mode.

On all commands that require a numeric value, the number is treated as follows:

- If the number contains an eight or nine, decimal base is assumed.
- The base may be specified by a post radix of B or D, that is, 562D is treated as 562 decimal and 562B is treated as 562 octal.
- Default conversion of numbers is normally to base octal although this is a system option.

TERMINAL CONTROL COMMANDS

The terminal control commands allow the user to vary the source and format of information given to and received from the system. These commands, which may be entered at any time after the user has successfully logged in, are described as follows:

ASCII	Provides capability to use the full ASCII or correspondence code set. This command specifies that subsequent characters entered from the terminal will be translated into an extended internal character set consisting, on an ASCII terminal, of the 96 graphic characters of the center four columns of the ASCII character set. On a correspondence code terminal, the extended internal character set normally represents all graphics defined for that terminal and character set. The

† The backspace character on Teletype model 33/35 terminals is the shift-Ø key. The character may be either an underline (_) or a left arrow (←).

standard internal character set contains only the first 63 or 64 or these characters, depending upon terminal mode and installation option. Character sets and a technique to determine which character set is installed are given in appendix B. If the terminal type is TTYD, MEMAPLD, CORAPLD, CORD, or BLKEDT, only the first 63 characters are available.

KRONOS recognizes all characters of the full ASCII code set except LINE FEED and the ASCII-DEL codes. Input control characters (return, delete, and backspace) are recognized but are not translated. Characters of the standard code set are processed internally as six-bit display code characters. The additional characters which make up the full ASCII code set are processed internally as 12-bit display code characters with an escape code convention. Refer to appendix B for an explanation of how the system interprets characters in both ASCII and non-ASCII modes.

The ASCII command must be entered if lowercase letters are to be interpreted as such by the system. Normally, all lowercase letters are translated to uppercase. Commands may be entered in uppercase or lowercase in either ASCII or non-ASCII mode.

NOTE

If a correspondence code terminal is being used, the ASCII command must be entered to enable use of the full correspondence code set.

AUTO,nnnnn,iiii

Causes generation of five-digit line numbers automatically. The nnnnn parameter specifies the beginning line number; default value is 00100. The iiii parameter specifies the increment value added for each succeeding line number; default value is 10. The user can exit from auto mode by using either of the following methods.

1. Delete the current line by pressing the ESC or ESCAPE key (ASCII code compatible terminals) or ATTN key (correspondence code terminals) and then enter a new command on the next line.
2. Backspace six character positions and then enter a new command on the same line (six backspaces are required to overwrite the five-digit line number and the blank that follows).

It is also possible to alter the line numbering sequence by deleting or backspacing (steps 1 or 2) and then entering a new beginning line number rather than a new command (leading zeros are permissible but not required). The user should exercise caution when doing this since the AUTO command is still in effect and continues generating line numbers using the original increment value. Thus, if a line number is generated that already exists in the file, the original content of that line is lost and must be re-entered. Note that it is not possible to alter the increment value unless a new AUTO command is entered.

In the following example of the AUTO command, the user backspaces over the line number and enters an LNH command (refer to Time-Sharing Job Commands in this section for a description of the LNH command).

```
AUTO
00100 THIS IS A TEST
00110 FOR THE AUTO COMMAND
00120 END
00130_______LNH

00100 THIS IS A TEST
00110 FOR THE AUTO COMMAND
00120 END
READY.
```

In the following example, the user backspaces to change the numbering sequence and later deletes a line to change the numbering sequence.

```
NEW,KL
READY.
AUTO,10,10
00010 THIS IS A TEST
00020 FOR THE AUTO COMMA ND
00030_______00100 TEST
00110 TEST
00120  *DEL*
00200 TEST
00210 END
00220  *DEL*
LNH

00010 THIS IS A TEST
00020 FOR THE AUTO COMMAND
00100 TEST
00110 TEST
00200 TEST
00210 END
READY.
```

CSET, c

Sets the terminal mode to the specified mode.

c	Specifies terminal mode	
	ASCII	Set ASCII mode
	NORMAL	Set NORMAL (not ASCII) mode

The CSET, ASCII, CSET NORMAL, and ASCII commands may also be entered through control cards included in a procedure file. The CSET command only affects the terminal character set, not its AUTO, PARITY, or TAPE states.

FULL

Enters full-duplex mode. Under this mode, each character received by the system is echoed back to the terminal just as it was received. This mode is effective only for terminals which have a full-duplex capability. The system responds:

```
READY.
```

If the half/full duplex switch on the terminal is in the half-duplex position when this command is entered, each subsequent character entered is double printed (initially when it is typed, and again when it is echoed back by the system). Placing this switch in the full-duplex position allows only the character echoed back to the terminal to be printed.

HALF

Clears full-duplex mode (refer to FULL command). Characters received by the system after this command is issued are not echoed back to the terminal. The system responds:

READY.

If the terminal being used has a half/full duplex switch, characters entered at the terminal are not printed when the switch is in the full-duplex position. Only system-generated output appears at the terminal. Placing this switch in the half-duplex position enables keyboard entry to be printed.

NORMAL

Reverses the effect of the ASCII, AUTO, PARITY, and TAPE commands on both input and output. The system initially assumes that this command has been entered. Normal mode consists of a 64-character code set where all lowercase letters are converted to uppercase. †

PARITY, p

Set terminal parity where p is either:

ODD to set odd parity
EVEN to set even parity

If no parameters are supplied, ODD parity is assumed.

The system initially assumes that all information is transmitted in even parity to ASCII code terminals and odd parity to correspondence code terminals.

ROUT, nn

Sets the amount of time required to perform the return function (carriage return and line feed) for an ASCII code terminal. This command is not valid for correspondence code terminals. The nn parameter specifies a character count delay. This is the amount of time required by the system to send nn characters to the terminal, where the value of nn can range from 0 to 30. If zero is specified, a standard delay value is used. The system responds:

READY.

This command is necessary since the length of time required for the return operation varies depending upon the type of terminal being used. For example, if the system is transmitting output to a terminal, a fixed amount of time is allowed for the return function, after which the system sends the next line of output. If the amount of time is not sufficient, characters may be printed during the carriage return function.

The delay is accomplished by sending rubout characters for paper tape operations and null characters for all other input/output operations. The number of characters to be sent is initially set at log-in or recovery time by the system. If a ROUT command is entered, the number of characters specified for the delay remain in effect until termination of the session (log-off), recovery, or until another ROUT command is entered.

† or 63, depending upon installation option.

TAPE

Permits subsequent information to be read from the paper tape reader at an ASCII code terminal. The system sends an X-ON character to the terminal at the end of each program request for data and after execution of each command. The X-ON character is used to turn on the paper tape reader at the user's terminal. This mode also inhibits the output of header messages from the LIST command and the READY message after the execution of most commands. The user should note that the system ignores carriage return characters entered on an empty input line while in tape mode. Refer to appendix C for complete information concerning paper tape operations.

TERM, t

Allows the user to redefine the terminal characteristics established at log-in. The t parameter specifies the new terminal characteristics as follows:

TTY	ASCII code terminal with standard print
TTYD	As TTY but with 63-character set
COR	Correspondence code terminal with standard print
CORD	As COR but with 63-character set
CORAPL	Correspondence code terminal with APL print
CORAPLD	As CORAPL but with 63-character set
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print
MEMAPLD	As MEMAPL but with 63-character set
BLKEDT	Block transmission terminal with full display screen capability (available only on certain terminals such as Hazeltine 2000)
BLKEDTD	As BLKEDT but with 63-character set

The TTYD, MEMAPLD, CORAPLD, CORD, and BLKEDTD translation tables are provided for compatability with previous systems.

The user can change the print type and then indicate the change by means of the TERM command. For example, a user operating a Memorex 1240 terminal changes the print type from standard to APL. To notify the system, enter:

TERM, MEMAPL (CR)

This command may be selected by default. This can be checked by entering the LIMITS command.

TIMEOUT

Changes a no timeout terminal to the standard timeout status. In standard timeout status, the user is automatically logged-off after 10 minutes of no action. Standard status is in effect when bit 11 in access word AW is clear. (Refer to the LIMITS command in this section.) When the bit is set, the terminal remains connected until the user logs off. The TIMEOUT command clears this bit for the session in progress.

This command is useful only for terminals capable of having either standard or APL print. The user can change the print type and then indicate the change by means of the TERM command.

For example, a user operating a Memorex 1240 terminal changes the print type from standard to APL. To notify the system, he enters:

TERM, MEMAPL (CR)

NOTE

All ASCII code terminals with standard print are designated TTY. The Memorex 1240 is an ASCII code terminal and therefore is identified as TTY when using standard print.

SUBSYSTEM SELECTION COMMANDS

The following commands are used to select a specific subsystem. The user should always be aware of the subsystem that is currently active. For example, attempting to execute a FORTRAN program while operating under the BASIC subsystem causes meaningless diagnostic messages to be issued. To determine which subsystem is currently active, enter the ENQUIRE command.

When a user enters the SAVE command to retain his primary file in the permanent file system, it is especially important that he is aware of which subsystem is currently active. An internal indicator called the subsystem flag associates the current subsystem with the primary file being saved. Thus, each subsequent request for that file using the OLD command causes the associated subsystem to be selected automatically. To save the primary file with no subsystem association, the user must enter the NULL subsystem before he issues the SAVE command. Note that the subsystem flag has meaning only when saving the primary file. When saving working files other than the primary file, no subsystem association is made.

BASIC, ccc

Selects the BASIC subsystem. The ccc parameter is optional and enables the user to specify an additional command. Although any valid command is permitted (as well as all valid parameters for that command), only one additional command may be specified. The following example illustrates the use of this option.

BASIC, OLD, PRIME

In this example, the user selects the BASIC subsystem and the OLD file PRIME in one command. Note that the file name (PRIME) is a valid parameter with the OLD command.

If ccc is omitted, and no primary file is currently defined, the system responds:

OLD, NEW, OR LIB FILE:†

The user may then specify the file type and the file name (separated by a comma) followed by a (CR) . If the file name is omitted in the reply, the system responds:

FILE NAME:

The user then specifies the file name followed by a (CR) .

† LIB is an abbreviation of the command LIBRARY. Refer to the description of this command later in this section.

BATCH, nnnnn

Selects the BATCH subsystem. The nnnnn parameter specifies the field length in octal central memory words required for control card images to be entered. If nnnnn is not specified, a default value of 20000B is assumed. The system responds:

```
$RFL, nnnnn.                    (nnnnn is the specified field length.)
/
```

The user may then enter control card images, time-sharing commands, or both for the job. Refer to the KRONOS Reference Manual for information concerning control cards.

It is important to note that the time-sharing commands and control card images entered are not processed as a batch job. as the subsystem name implies. Instead, each control card or command is entered and processed individually. The subsystem is named BATCH because images of control cards, usually referred to as batch control cards, are able to be processed. Refer to section 9 for a complete description of the BATCH subsystem.

EXECUTE, ccc

Selects the EXECUTE subsystem. This subsystem is used only to execute a previously compiled (object code) program. The RUN (or RNH) command must be entered to initiate execution. The ccc parameter is optional and enables the user to specify an additional command. Although any valid command is permitted (as well as all valid parameters for that command), only one additional command may be specified. The following example illustrates the use of this option.

```
EXECUTE,OLD,OBJFILE
```

In this example, the user selects the EXECUTE subsystem and the OLD file OBJFILE in one command. Note that the file name (OBJFILE) is a valid parameter with the OLD command.

If ccc is omitted and no primary file is currently defined, the system responds:

```
OLD, NEW, OR LIB FILE:†
```

The user may then specify the file type and the file name (separated by a comma) followed by a (CR) . If the file name is omitted in the reply, the system responds:

```
FILE NAME:
```

The user then specifies the file name followed by a (CR) .

NOTE

Source language programs cannot be executed under the EXECUTE subsystem.

The EXECUTE subsystem should be used whenever possible to improve time-sharing efficiency. This can be achieved when certain programs are used frequently. For example, a source code program created under the FORTRAN subsystem must first be compiled before it can be executed. This process is automatic. The user needs only to enter the RUN command.

†LIB is an abbreviation of the command LIBRARY. Refer to the description of this command later in this section.

The program automatically goes through a compilation phase which produces an executable object code (binary) program. The object code program is then executed. From this, it can be seen that retaining the object code program in the permanent file system for later execution under the EXECUTE subsystem provides greater efficiency. This can be accomplished through use of the RUN,B=lfn command. This command causes the object code program to be placed on a working file (lfn) specified by the user. That file is then retained in the permanent file system by entering the command SAVE,lfn.

Note that since the file containing the object code is not the primary file, a subsystem flag is not associated with the file when it is saved. In this case, the EXECUTE subsystem will not be selected automatically when the file is requested using the OLD command. The user must ensure that the EXECUTE subsystem has been selected before attempting to execute the program.

To associate the EXECUTE subsystem with the file containing the object code program, enter the following commands.

EXECUTE,NEW, lfn,ND	Selects the EXECUTE subsystem and specifies a new primary file name (lfn). The ND parameter prevents the working file containing the object code program from being released
SAVE	Retains the primary file in the permanent file system and sets the EXECUTE subsystem flag. The file contains no information at this point.
REPLACE,lfn=pfn	Copies the contents of the working file containing the object code program (lfn) to the permanent file created with the SAVE command (pfn). This command can be entered regardless of the subsystem currently being used. It has no effect on the EXECUTE subsystem flag associated with file pfn.

If this is done, subsequent requests to the file using the OLD command cause the EXECUTE subsystem to be selected. Entering the RUN command initiates execution of the object code program.

FTNTS,ccc

Selects the FORTRAN Extended subsystem. The ccc parameter is optional and enables the user to specify an additional command. Although any valid command is permitted (as well as all valid parameters for that command), only one additional command may be specified. The following example illustrates the use of this option.

```
FTNTS,OLD,DATAGEN
```

In this example, the user selects the FORTRAN extended subsystem and the OLD file DATAGEN in one command. Note that the file name DATAGEN is a valid parameter with the OLD command.

If ccc is omitted and no primary file is currently defined, the system responds:

OLD, NEW, OR LIB FILE:†

The user may then specify the file type and file name (separated by a comma) followed by a CR . If the file name is omitted in the reply, the system responds:

FILE NAME:

The user then specifies the file name followed by a CR .

NULL — Clears the subsystem currently in use. The system responds:

READY.

This command is entered before saving the primary file if the user does not want a specific subsystem flag to be associated with the file. Normally, when a user saves the primary file, an internal indicator called the subsystem flag is set to indicate the subsystem currently in use. That subsystem is automatically selected in each succeeding request for the file using the OLD command. Note that there is no subsystem association when saving working files other than the primary files.

If the user does not specify a subsystem at log-in time, the system assumes NULL by default.

TIME–SHARING JOB COMMANDS

The time-sharing job commands allow the user to perform a variety of job processing or job related operations. These commands may be entered at any time after the log-in procedure is complete.

APL,t — Selects the APL (A Programming Language) interactive interpreter. This command also specifies translation to the full 128-character code set. All characters are recognized although input control characters are not processed. Rather, they are passed to the program as data. The t parameter is optional and identifies the type of terminal being used. Refer to the KRONOS APL*CYBER Reference Manual for a list of terminal identifiers and information concerning the use of APL.

† LIB is an abbreviation of the command LIBRARY. Refer to the description of this command later in this section.

BINARY,nnnn,ccc

Selects binary input mode. Under control of this command, eight-bit characters entered from the terminal are stored, without translation, in the primary file. Normally, characters entered from the terminal are translated and stored as 6-bit or 12-bit internal display codes (refer to Central Memory Character Equivalents in appendix B for complete information).

nnnn Number of characters (1 to 4095) to be entered

ccc Termination character; a three-digit octal code that corresponds to a character which the user selects to terminate binary input mode. Refer to appendix B for the correct character set and corresponding codes. (If a correspondence code terminal is being used, a value of 200 must be added to the octal code listed in appendix B.)

The termination character enables the user to return to the normal input mode before the number of characters specified by nnnn has been entered.

Example:

BINARY,20,207 (CR)

In this example, binary input mode is terminated when 20 characters have been entered or the ASCII character BELL (corresponds to octal code 207 entered in command) is typed.

NOTE

If the termination character (ccc) is omitted, the number of characters specified by nnnn must be entered to terminate binary input mode.

BYE

Logs the user off the system and disconnects the terminal. The system responds by printing:

usernum LOG OFF. hh.mm.ss.

usernum SRU s.sss UNTS.

Where:

usernum User number

hh.mm.ss. Time of day this command was entered

s.sss Total SRUs used under current charge/project number

NOTE

The user should make certain that all working files to be retained are made permanent before issuing this command. All working files not saved are released when the BYE command is processed.

CHARGE, chargeno, projectno

Allows the user to specify a valid charge number and project number to be billed for subsequent terminal operations. Although certain users are required to enter a charge and project number during the log-in sequence, all users may issue the CHARGE command as long as the charge and project numbers entered are valid. This command can be entered at any time during the time-sharing session. Both charge number (chargeno) and project number (projectno) parameters are required. The SRU accumulator and usage summary is written to the user's dayfile and the SRU accumulator is set to zero (0). The SRU is a unit of measurement which represents the usage of the system by the user. The SRU represents all CPU time, memory usage, and I/O activity. The system responds:

```
READY.
```

CLEAR

Releases all working files. If a primary file is currently specified, only the name of the primary file is retained. The system responds to entry of the CLEAR command with:

```
READY.
```

All files to be retained must be made permanent before entering this command.

CONVERT, p_1, p_2, . . . , p_n

Converts records from one character set to another.

p may be any of the following in any order:

P=lfn_1	Input on file lfn_1. If omitted, input on file OLD.
N=lfn_2	Output on file lfn_2. If omitted, input on file NEW.
RS=n	Maximum record size in characters, where n is from 1 to 500_{10}. If omitted, assume maximum record size is 300_{10} characters.
64	Convert from 63- to 64-character set. If
64 omitted	omitted, no 63- to 64-character set conversion (TS must be specified if 64 is not).
TS=t	Convert from old to new time-sharing character set. t may be one of the following terminal types:

t	Terminal Type
TTY	ASCII code terminal with standard print
COR	Correspondence code terminal with standard print
CORAPL	Correspondence code terminal with APL print
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print
BLKEDT	ASCII code terminal with standard print, block edit mode

If t is omitted, it is assumed to be TTY. If TS is omitted, no time-sharing conversion; 64 must be specified if TS is not.

R — Rewind I/O files prior to processing. If omitted, no rewinding occurs.

RC=n — Convert n decimal records. If n is omitted, convert to EOF. If RC is omitted, convert one record.

The following conversions are possible as indicated by the legal choice of parameters.

Type of Record	Legal Conversion
63-character set, non-time-sharing record	64
Old time-sharing record	TS or 64 and TS
New NORMAL time-sharing record (equivalent to BATCH character set)	64
New ASCII time-sharing record	None

The following messages may be issued to the user's dayfile in response to a CONVERT control card.

DAYFILE — Prints the system's record of the user's time-sharing activity. Each line of output returned to the terminal is in the form

hh.mm.ss. message

Where:

hh.mm.ss. Time of day the message was placed in the dayfile

EDIT, lfn, AS — Selects the Text Editor. The lfn parameter specifies the name of an existing working file to be edited. The lfn parameter may also specify a new working file name. In this case, the new file is constructed using the Text Editor. If the lfn parameter is omitted, the primary file name is assumed. When necessary, the primary file is sorted prior to the initiation of text editing.

AS parameter indicates that the user wishes to process the file in ASCII mode. The terminal is automatically put in ASCII mode. After editing is completed, the terminal is put back in its original mode. If the user is in ASCII mode before the Text Editor is entered, the *AS* parameter is assumed present.

All working files consisting of several logical records should be compressed into one logical record before entering this command (refer to PACK command described in this section). If this is not done, the contents of all but the first logical record is lost. For additional information, refer to the Text Editor (EDIT) Reference Manual.

ENQUIRE

Gives information about the system to the user. Three formats of the command are allowed.

- ENQUIRE(OP=ppppppp, JN=JOBNAME, FN=FILENM2, 0-FILENM1)

  ppppppp is one of the options listed below.

- ENQUIRE, X — X is any valid option. For example, ENQUIRE, BFS selects options B, F, and S.

- ENQUIRE — Gives the current job status. This version of the command is given after the options are described.

Valid Options:

OP=A, Causes all the OP= options to be executed

OP=B, Returns user identification information and priority such as USER NUMBER, FAMILY NAME, CPU PRIORITIES, and USER INDEX to the user.

OP=F, Status for files at user's control point

For example:

FILENAME	LENGTH/PRUS	TYPE	STATUS
BFILEB	81	LR.	EOR
OUTPUT	7	PR.	IC-R/W
OPL	19349	PM.	BOT

OP=J, Returns the contents of control registers and error flag field to the user

OP=L, Returns loader information to the user

OP=R, Returns the amount of resources used (CPU time, MS activity, MT activity, etc.) to the user

OP=S, Gives the user his SRU. This unit represents the total usage of the system by the user and is derived from CPU time, I/O activity, and memory usage.

OP=T, Returns the accumulated CPU time for this session. For example:

CPU TIME 20.46 seconds

OP=U, Returns the amount of resource usage allowed to the user (dispose files, mass storage, dayfile messages, and control statements)

JN= jobname — Returns the status of a remote batch job that was initiated with the SUBMIT command. The jobname parameter is the name assigned to the job, by the system, when it is submitted. The possible replies are:

JOB NOT FOUND.
JOB IN INPUT QUEUE.
JOB IN OUTPUT QUEUE.
JOB IN PUNCH QUEUE.
JOB EXECUTING.

The JOB NOT FOUND. message normally indicates that the job has been processed and no longer exists in the system. However, if the jobname is entered incorrectly (misspelled), this message may also be issued.

The JOB EXECUTING. message indicates that the job is either executing or has been moved out for a higher priority job. Only the status of jobs submitted under the current user number can be obtained.

FN= filenm2 — Returns the same information as the F option about a particular file. The file must be local to the user control point. The following message is given if the file is misspelled or is not at the user's control point.

FILE NOT FOUND.

The filenm2 is the name of the file to find.

O= filenm1 — Requests output to be placed on an alternate file. The filenm1 parameter is the name of the alternate file. Default name for filenm1 is output.

ENQUIRE — Requests the current job status. An important feature of this command is that it may be entered during job execution. The system responds by printing:

TERMINAL	nnn, iii
SYSTEM	(subsystem currently active)
FILE NAME	
STATUS	IDLE — No job being processed
	EXECUTE — User job being processed
	WAIT — User job waiting for computer resources to become available

Where:

nnn — Terminal number

iii — Identifier that describes type of terminal

TTY	ASCII code terminal
COR	Correspondence code terminal
CORAPL	Correspondence code terminal with APL print

MEMAPL	Memorex 1240 (ASCII code) terminal with APL print
BLKEDIT	Block transmission terminal

GOODBYE — Same as BYE command

HELLO — Logs the current user off the system, as in the BYE command, and reinitiates the log-in sequence. Any working files that are to be retained must be made permanent before the HELLO command is entered. This command allows a new user to log-in without first having to establish communications with the KRONOS system by dialing the computer site.

HELP — Allows the time-sharing user to obtain assistance in the use of KRONOS commands. If the user at a terminal is not sure of the format of a command or its use, the HELP command provides an easily accessible on-line description of the KRONOS command set. The system responds by printing:

```
IF MORE INFORMATION NEEDED, TYPE YES. OTHER-
WISE, TYPE THE COMMAND NEEDED.
?
```

The user then types YES in reply to the request or else enters the command he wishes to have described. A minimum of three characters must be entered if a command is typed. The user must type END after a question mark (?) to terminate the HELP command.

I key — Interrupts a program that is currently transmitting output to the terminal. The system responds:

```
*INTERRUPTED*
```

The user must press the CR key or type P followed by CR to continue. However, if the TAPE command is in effect (tape mode), CR cannot be entered as the first character of a new line. In this case, P CR must be entered to continue. Refer to Job Suspension in section 7 for additional information.

Use of the I key to perform this function is effective only for ASCII code terminals. The ATTN key can be used to perform the same function on correspondence code terminals (refer to Table B-1 in appendix B for complete information concerning the use of special function keys for both ASCII and correspondence code terminals).

If pressing the I key accidentally disconnects the terminal from the system, the user can log-in again and enter the RECOVER command in response to the RECOVER/SYSTEM: or RECOVER/CHARGE: request. He can then continue from the point at which he was disconnected. Refer to Recovery in section 7 for additional information.

LENGTH, lfn — Prints the file name, file type, last status, and length of the working file specified by lfn. If the lfn parameter is omitted, the primary file is assumed. In this case, the primary file is sorted before the length is obtained, unless a NOSORT command precedes the LENGTH command (refer to description of NOSORT command in this section).

The system responds with:

LOCAL FILE INFORMATION

FILENAME	LENGTH/PRUS	TYPE	STATUS
lfn	lllllll	tt	sss

lfn — Local file name

tt — File type

PM.	Direct access permanent file
PT.	Primary working file
LO.	Any working file that is not the current primary file or a direct access permanent file
CM.	Common file
LI.	Library file
SY.	System file

A * preceding the file type indicates a locked file (user cannot write on a locked file).

sss — File status

EOI	File is positioned at end-of-information.
EOF	File is positioned at end-of-file.
EOR	File is positioned at end-of-record.
I/C	File is positioned somewhere between beginning and end of information.

lllllll — File length in PRUs. (Each PRU contains 640 six-bit characters.)

LIBRARY — Allows a user to access a file that was previously saved in the permanent file system under special user number LIBRARY. The system responds:

FILE NAME:

The user then enters the selected file name. When the file is found, the system replies:

READY.

The file name specified becomes the new primary file and the previous primary file is released. All other current working files are also released unless the ND parameter is specified (refer to section 5).

LIBRARY,lfn=pfn/options — This is an optional form of the preceding command. It allows the user to specify the command LIBRARY, the selected file name, and special options on the same line. Refer to the command description in section 5 for information concerning this command format. Special considerations in the use of this command are described separately at the end of section 5.

LIMITS — Provides a listing of the user's current validation limits. Refer to Limits Command at the end of this section for a complete description of this command and a sample listing.

LIST,xxxxx,R,
F=lfn

Prints the contents of the primary file at the terminal unless the F option is specified. The primary file is automatically sorted before listing unless a NOSORT command precedes this command (refer to description of NOSORT command in this section).

xxxxx — Line number of the line where listing is to begin. This parameter is optional and may be specified only if the primary file is being listed. The NOSORT command should precede the LIST command if this option is specified since the file should be sorted to obtain correct results.

R — End-of-record and end-of-file marks are to be indicated in the listing if present (optional). This lists the primary file from the beginning to the end-of-information unless the xxxxx parameter is specified.

F=lfn — Lists working file lfn (optional). The file is not sorted before listing. When this parameter is specified, neither the xxxxx nor the R option can be supplied. The working file specified is listed from the current position to the end-of-information.

The system replies:

```
yy/mm/dd.     hh.mm.ss.   }
PROGRAM  nnnnnnn          }  header
     .
     .
(program lines)
     .
     .
READY.
```

The header information printed is:

yy/mm/dd. — Current date

hh.mm.ss. — Current time

nnnnnnn — File name

LNH,xxxxx,R,
F=lfn

This is an optional form of the preceding command which lists the file without a header.

LOGIN

Same as HELLO command

NEW

Allows the user to create a new primary file. The system responds:

FILE NAME:

The user enters a valid file name. When the system accepts the new file name, it replies:

READY.

The file name specified becomes the new primary file and the previous primary file is released. All other current working files are also released unless the ND parameter is used (refer to section 5).

NEW,lfn

This is an optional form of the preceding command which allows the user to specify NEW and the file name in one step.

No Drop

The /ND parameter is used to prevent dropping of working files (refer to section 5 for its use with other commands).

NOSORT

Prevents the system from automatically sorting the primary file. This is done by clearing an internal indicator called the sort flag. Sorting of the primary file is initiated only if the sort flag is set when the LIST, RUN, EDIT, LENGTH, or SUBMIT command is entered. The system responds:

READY.

This command is generally used when making additions and/or modifications to the primary file. When followed by the NOSORT command, the additions and/or modifications create a new logical record in the primary file. Note that the NOSORT command remains in effect only until the next numbered line of source code is entered. This causes the sort flag to be turned on again. Refer to the description of file sorting in section 3 for an example which illustrates the effect of the NOSORT command.

OLD

Allows the user to access a file that was previously saved in the permanent file system. The system responds:

FILE NAME:

The user then enters the selected file name. When the file is found, the system replies:

READY.

The selected file becomes the new primary file. If a specific subsystem is associated with the file, it is selected automatically. This occurs only if the file was originally the primary file and was saved while a subsystem other than NULL was active. All working files are released following entry of this command unless the ND parameter is used (refer to section 5).

OLD,lfn=pfn/options

This is a special form of the preceding command. It allows the user to specify the command OLD, the file name, and special options on the same line. Refer to the command description in section 5 for information concerning this command format.

PACK,lfn_1,lfn_2,NR

Compresses (that is, packs) a file consisting of several logical records into one logical record. When the operation is complete, the system responds:

READY.

The PACK command may be entered in one of the four following formats depending upon the options selected.

PACK

Packs the primary file into one logical record. When the primary file is packed, it is not sorted automatically by subsequent operations that normally cause the file to be sorted (that is, LIST, RUN, etc.). Automatic sorting resumes when the user enters the next line of source code or the SORT command.

PACK,lfn	Rewinds and packs working file lfn into one logical record. File lfn is rewound again after the pack.
PACK,lfn_1, lfn_2	Rewinds and packs working file lfn_1 into one logical record and then writes the file to lfn_2. If lfn_2 currently exists, lfn_1 is written at the current position of lfn_2. File lfn_2 is rewound after the pack; lfn_1 is not.
PACK,lfn_1, lfn_2,NR	Same as the preceding format, except that the no rewind (NR) parameter inhibits file lfn_1 from being rewound before the pack, unless it is the primary file. The primary file is rewound after all operations.

PASSWOR, oldpswd, newpswd

Allows validated users to change their password. The user must supply the correct current password (oldpswd) and a one- to seven-character, alphanumeric, new password (newpswd).

If any part of this command is in error, the system replies:

ERROR IN PASSWORD ARGUMENTS.

After the password has been changed, the system replies:

READY.

If the user is not validated to use this command, the system responds:

ILLEGAL CONTROL CARD.

PRIMARY, lfn.

Changes the primary file from the current primary file to the local file specified by lfn. The current primary file is dropped.

RECOVER

Enables the time-sharing terminal user to resume processing after having been accidentally disconnected from the system or when a system malfunction requires that the log-in sequence be reinitiated. Refer to Recovery in section 7 for a complete description and examples of the use of this command.

RENAME, lfn_1= lfn_2

Changes the name of working file lfn_2 to file name lfn_1. If file lfn_1 currently exists, it is released. This command cannot be used to change the name of a permanent file. However, a working file can be renamed and then saved in the permanent file system under the new name.

RESEQ

This command enables the user to resequence the line numbers of the primary file. If the current primary file is a text file, line numbers can be added. Statements that reference line numbers are updated when BASIC programs are resequenced. Refer to Resequencing Line Numbers in section 7 for a complete description and examples of the use of this command.

RETURN, lfn — Releases working file lfn

REWIND, lfn — Positions working file lfn at the beginning-of-information (BOI). Refer to File Structure in section 3 for additional information.

RUN, options — Compiles and/or initiates execution of the primary file or another working file. The primary file is sorted before being compiled unless a NOSORT command precedes this command (refer to description of NOSORT command in this section). The system responds:

```
yy/mm/dd.    hh.mm.ss.   }
PROGRAM      nnnnnnn     }  header
        .
        .
(data or error messages)
        .
        .
SRU          s.sss UNTS.
RUN COMPLETE.
```

One or more of the following options may be supplied with the RUN command.

RUN, B=lfn or RUN, C=lfn — Compiles the source program and generates a binary file lfn containing the resultant object code on file lfn. The source program is not executed. File lfn can later be executed under the EXECUTE subsystem by entering the RUN command. File lfn can be retained as a permanent file using the SAVE (or REPLACE) command.

RUN, I=lfn — Compiles and/or initiates execution of the working file lfn (lfn is assumed to be sorted). In the EXECUTE subsystem, no compilation occurs.

RUN, MA=nnnnn — Uses nnnnn as the octal field length for the current job †

RUN, MI=nnnnn — Increments the job's current field length by nnnnn octal. Refer to example 1. †

RUN, T, q_1, q_2, ..., q_n — This command applies only when running a previously compiled (object code) FORTRAN Extended program in the EXECUTE subsystem. It allows the user to rename working files specified in the PROGRAM statement without recompiling the program. The parameters q_1 through q_n are used to specify new working file names to be used in place of those that currently exist in the PROGRAM statement (the parameters are order-dependent and must correspond to those they replace). Refer to example 2.

† If the user exceeds the maximum field length for which he is validated, the message ILLEGAL USER ACCESS is returned to the terminal.

Example 1:

The following example illustrates the use of the RUN, MI=nnnnn command.

```
LNH

00100 PROGRAM M (INPUT,OUTPUT)
00110 DIMENSION J(17000) <-------------- Causes program to exceed
00120 PRINT 4                            field length (included solely
00130 4 FORMAT (*ENTER TWO NUMBERS*)     for purposes of illustration)
00140 READ,A,B
00150 C=A*B
00160 PRINT 5
00170 5 FORMAT (*A X B*)
00180 PRINT,C
00190 END
READY.

RUN

 75/03/01.  16.07.31.
PROGRAM  M

AT LINE NUMBER 00190 MORE STORAGE NEEDED FOR COMPILATION
000001  FORTRAN  ERRORS  IN  M                    |
                                                  Diagnostic indicates that job's
SRU     0.164 UNTS.                               current field length was in-
                                                  sufficient.
RUN COMPLETE.

RUN,MI=3000 <---------------------------- Additional field length is
                                          requested.
 75/03/01.  16.08.41.
PROGRAM  M

ENTER TWO NUMBERS
? 16,23
A X B
              368.00
 END.

SRU     0.396 UNTS.

RUN COMPLETE.
```

Example 2:

The following example illustrates the use of the RUN, T command. Assume a FORTRAN Extended program contains the following PROGRAM statement:

```
PROGRAM TEST (INPUT,OUTPUT,AAA,BBB,TAPE1=AAA,TAPE2=BBB)
```

Normally, to execute a program containing this statement, the user would have to define working file AAA. Working file BBB would result from the execution of the program. However, the user could change the names of working files AAA and BBB without recompiling the program (if the object program exists) by using the RUN,T command under the EXECUTE subsystem. For example:

```
RUN,T,INPUT,OUTPUT,CCC,DDD
```

RNH

This is an optional form of the RUN command which allows the user to run the job without a header being printed. The commands are identical in all other respects.

S key

The user presses this key to terminate a job that is currently transmitting output to the terminal. The use of this key is effective only for ASCII code compatible terminals. The system responds:

TERMINATED.

If pressing this key accidentally disconnects the terminal from the system, dial the KRONOS site and enter the RECOVER command in response to the RECOVER/SYSTEM: or RECOVER/CHARGE: request during the log-in sequence.

To terminate a job currently transmitting output to a correspondence code terminal, press the ATTN key. The system responds:

INTERRUPTED.

The user must then enter the STOP command which terminates the job and prints the message:

TERMINATED.

Refer to Table B-1 in appendix B for complete information concerning the use of special function keys for both ASCII and correspondence code terminals.

SETTL,nnnnn

Sets time limit. This command allows the user to specify a time limit (nnnnn octal seconds) for all subsequent operations requiring the central processor. If nnnnn is omitted, the system assumes nnnnn=100B seconds which is the normal default value at log-in time. The minimum time limit that can be entered is 10B seconds. The maximum time limit is dependent upon the user's validation limits (refer to Limits Command at the end of this section).

This command is useful when debugging a program that is cycling in a loop. It prevents the user from accumulating a large bill for computer usage.

SORT

Sets an internal indicator called the sort flag. Sorting of the primary file takes place when the next LIST, LENGTH, EDIT, SUBMIT, or RUN command is entered. The lines of the primary file are then arranged in order according to the first five digits of the line number, and the file is packed into one logical record.

SORT,lfn,
NC=n

Sorts working file lfn by arranging the statements in order according to the first n digits of the line number (where $1 \leq n \leq 10$). If the NC option is omitted, the sort is performed using the first five digits of the line number. File lfn is also packed into one logical record when the sort is performed. When the sort is complete, file lfn is positioned at end-of-information (EOI) unless it is the primary file which is automatically rewound. This command forces immediate sorting of the specified file and is commonly used to sort the primary file when it is too long to be sorted automatically by the system. Refer to description of File Sorting in section 3 for additonal information.

STATUS

Identical to the ENQUIRE (CR) command (no options allowed).

STOP

Terminates any program that is currently in execution or waiting for input from the terminal, unless the program has disabled terminal control (for example, APL). The system responds:

TERMINATED.

SUBMIT

Allows a validated user to create a batch job deck image and submit it to the local batch queue as a deferred batch job. The user cannot interact with a job submitted for processing using this command. Rather, the job is treated in the same manner as a job submitted from the card reader at the local computer site or from EXPORT/IMPORT. This process is commonly referred to as a remote job entry. Refer to section 8 for a complete description and examples of the use of this command. It is recommended that the user become familiar with all information in section 8 before attempting remote job entry.

TEXT

Enter text mode. This command allows keyboard entry of information (program statements, data, or text) into the primary file without specifying line numbers. Commands entered in text mode are interpreted as text and are not processed by the system. The system responds:

ENTER TEXT MODE.

To terminate text mode from an ASCII code terminal, press the interrupt key (generally labeled INT, INTRPT, or BREAK), or, if provided, the ETX (control=C) key. Since the interrupt key generates a NUL character (which is not recognized in tape mode), the ETX character must be used to terminate text mode when in tape mode.

If a correspondence code terminal is being used, press the ATTN key. Note that in either case, the carriage must be positioned at the beginning of a new line (empty input line) to be effective. The system responds:

EXIT TEXT MODE.

Since a text file does not normally contain line numbers, and therefore, cannot be sorted (and packed) automatically by the system, a PACK command should be issued after exit from text mode to ensure that the data is in one logical record.

TRAN, termnam

Selects the Transaction Subsystem (TRANEX). The termnam parameter specifies the name of the terminal. In this mode, the terminal can communicate with TRANEX. Operation of TRANEX is detailed in the TRANEX Reference Manual.

-lfn
or
CALL, lfn

Initiates processing of procedure file lfn. This allows the user to process a procedure file and remain interactive with his job.

A procedure file is a source file which contains control card images and/or control language statements. It has the same structure as the control card record in a normal batch job although the JOB and USER control cards required in batch jobs cannot be included.

Optional parameters may also be included in this command. The command format with options is:

-lfn,S=ccc(oldnam$_1$=newnam$_1$,oldnam$_2$=newnam$_2$....,
oldnam$_n$=newnam$_n$)

or

CALL,lfn,S=ccc(oldnam$_1$=newnam$_1$,oldnam$_2$=newnam$_2$,...,
oldnam$_n$=newnam$_n$)

Where:

lfn — Name of the procedure file to be processed. KRONOS searches for lfn according to the following hierarchy:

1. Check current working files (including primary file).
2. Check system-defined procedure files. This can be a procedure file created by the installation and included as part of the KRONOS system.
3. Check user's permanent file catalog for indirect access permanent file. File must not have an associated password.

If the specified file cannot be found, the system responds:

lfn NOT FOUND.

S=ccc — Specifies control statement ccc as the first statement in the procedure file to be executed

oldnam=newnam — For each occurrence of oldnam in the procedure file, substitute newnam. This allows use of the same procedure file to perform several tasks depending upon the parameters specified.

To create a procedure file from the terminal, the user should enter text mode (refer to TEXT command). After leaving text mode, issue the PACK command to ensure that the file is packed into one logical record. The file can then be SAVEd for future reference as an indirect access permanent file. If changes have to be made to the procedure file, it is necessary to use the Text Editor (refer to EDIT command).

Refer to the KRONOS Reference Manual for information concerning the control statements and control language statements available.

LIMITS COMMAND

Entry of the LIMITS command provides a listing of the user's validation limits. Generally, validation limits are the internal system controls associated with each user number which govern his use of certain system resources. The listing provided describes both the resources available to the user and the extent to which they may be used. All numeric values listed are decimal unless the post radix B appears, signifying an octal value. The following information is listed.

Field	Description
AB	Answerback identifier (1 to 10 alphanumeric characters). Terminals with answerback capability automatically send this identifier to the system during log-in. The system uses the information to determine if it is a legal terminal. Each user number can have up to four terminal answerback identifiers associated with it.
MT†	Maximum number of magnetic tapes the user is allowed to have assigned to his job concurrently
RP†	Maximum number of auxiliary devices the user is allowed to have assigned to his job concurrently
TL	Maximum amount of central processor time in seconds allowed for the user's job (refer to SETTL command)
PR†	Maximum central processor priority allowed for the user's job
CM†	Maximum number of central memory words that the user is allowed to request. This is the job's maximum field length and is expressed as multiples of 100B words.
NF††	Maximum number of files that the user is allowed to have associated with his job concurrently
MS††	Maximum number of mass storage tracks the user is allowed to use
PN††	Project number to which the user is assigned (not the project number associated with the charge number entered at log-in with the CHARGE command)
DB††	Maximum number of deferred batch jobs that the user can have in the system concurrently (refer to description of SUBMIT command in section 8).
FC	Maximum number of permanent files the user can have in each catalog. This limit applies for each catalog being accessed (main, public auxiliary, or private auxiliary catalogs).
CS	Maximum number of PRUs available to the user for indirect access permanent files.
FS	Maximum number of PRUs available to the user for any given indirect access permanent file.
PA	Specifies that ODD or EVEN parity is associated with the terminal (refer to PARITY command in Terminal Control Commands in this section).
RO	Number of rubout characters required for carriage return delay (refer to ROUT command in Terminal Control Commands in this section).

† For further information about this field, refer to the KRONOS Reference Manual.
†† Not currently used by the system but provided for future expansion

Field	Description
PX	Specifies full- or half-duplex transmission mode is associated with terminal (refer to FULL and HALF commands in Terminal Control Commands in this section).
TT	Specifies type of terminal; value listed may be:

TTY	ASCII code terminal with standard print
TTYD	As TTY but with 63-character set
COR	Correspondence code terminal with standard print
CORD	As COR but with 63-character set
CORAPL	Correspondence code terminal with APL print
CORAPLD	As CORAPL but with 63-character set
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print
MEMAPLD	As MEMAPL but with 63-character set
BLKEDT	Block transmission terminal with full display screen capability (available only on certain terminals such as Hazeltine 2000)
BLKEDTD	As BLKEDT but with 63-character set

(The TTYD, MEMAPLD, CORAPLD, CORD, and BLKEDTD translation tables are provided for compatability with previous systems.)

Identifier	Description
TC = xxxxx	xxxxx specifies the character set to be used by the terminal. One of the following values is available for each user number.

Value	Explanation
STANDARD	63- or 64-character code set (depending up installation option)
ASCII	128 full ASCII code set

IS = xxxxx

xxxxx specifies the initial subsystem for the terminal. One of the following values may be specified for each user number.

Value	Explanation
NULL	NULL subsystem
BASIC	BASIC subsystem

Value	Explanation
FTNTS	FORTRAN Extended subsystem†
TRANSACT	TRANSACT (TRANEX) subsystem †
EXECUTE	EXECUTE subsystem
BATCH	BATCH subsystem

Field	Description
PA	Specifies that ODD or EVEN parity is associated with the terminal (refer to PARITY command under Terminal Control Commands in this section).
RO††	Number of rubout characters required for carriage return delay (refer to ROUT command under Terminal Control Commands in this section).
PX	Specifies full- or half-duplex transmission mode is associated with terminal (refer to FULL and HALF commands under Terminal Control Commands in this section).
TT	Specifies type of terminal; value listed may be:

TTY	ASCII code terminal with standard print
MEMAPL	Memorex 1240 terminal (ASCII code with APL print)
COR	Correspondence code terminal with standard print
CORAPL	Correspondence code terminal with APL print
BLKEDIT	Block transmission terminal with full display screen editing capability

Refer to TERM command under Terminal Control Commands in this section.

MS	Maximum number of mass storage PRUs the user is allowed to additionally allocate to his job
DF	Maximum number of MESSAGE requests the user can issue to the system and/or job dayfiles
CC	Maximum number of batch control card statements processed for a user (excludes TELEX processed control statements)
OF	Maximum number of job print and punch files the user can dispose to the output queues
CP	Maximum number of cards that can be punched from a user's disposed punch file

†Refer to the TRANEX Reference Manual
††If the value listed for this field is SYSTEM, the actual value used is a system option.

Field	Description
LP	Maximum number of lines that can be printed from a user's disposed print file
EC†	Maximum number of ECS memory words that the user is allowed to request
SL†	Maximum number of SRUs allowed
CN	Charge number for this user
PN	Project number to which the user is assigned (not the project number associated with the charge number entered at log-in with the CHARGE command)
DS	Maximum number of PRUs available to the user for any given direct access permanent file
AW	Access word; the octal value listed corresponds to the following access options (bit 0 is option 1, bit 1 is option 2, etc.).

Option	Signifies
1	User can change his password (refer to PASSWOR command).
2	User can use privileged time-sharing commands (refer to appendix E in the KRONOS Operator's Guide).
3	User is allowed to create direct access permanent files.
4	User is allowed to create indirect access permanent files.
5	User can have system origin status from any job origin if the system console is in DEBUG mode (refer to KRONOS Reference Manual).
6	User can access common files (refer to KRONOS Reference Manual).
7	User can assign nonallocatable devices (refer to KRONOS Reference Manual).
8	User is allowed to access system without supplying his assigned charge and project numbers.
9	User can create and/or replace files on auxiliary devices.

† Not currently utilized by the system, but provided for future expansion of validation control.

Option	Signifies
10	User can access special transaction functions (refer to TRANEX Reference Manual).
11	User is designated as a no timeout terminal. If this bit is not set, the terminal will automatically log off after 10 minutes of inactivity. With this bit set, the terminal remains connected until the user logs off. The TIMEOUT command (section 4) clears this bit.

For example, if the access word listed were

AW=0000000000000000215

the user would be validated for options 1, 3, 4, and 8.

If any parameters are included with the LIMITS command, the system responds:

ERROR IN LIMITS ARGUMENTS.

The following is an example output listing produced by LIMITS.

```
        LIMITS.                          75/04/30. 09.39.27.          PAGE
DAVEHIN  64    74/10/03. 75/01/08.
      AB =,
      AB =,
      AB =,
      AB =,
    MT =  UNLIMITED,
    RP =  UNLIMITED,
    TL =  UNLIMITED,
    CM =  UNLIMITED,
    NF =  UNLIMITED,
    DB =  UNLIMITED,
    FC =      SYSTEM,
    CS =      SYSTEM,
    FS =      SYSTEM,
    PA = EVEN       ,
    RO =            0,
    PX = HALF       ,
    TT = TTY        ,
    TC = STANDARD   ,
    IS = NULL       ,
    MS =  UNLIMITED,
    DF =  UNLIMITED,
    CC =  UNLIMITED,
    OF =  UNLIMITED,
    CP =  UNLIMITED,
    LP =  UNLIMITED,
    EC =  UNLIMITED,
    SL =  UNLIMITED,
     CN =,
     PN =           ,
    DS =      SYSTEM,
    AW = 0000000000000003777
READY.
```

PERMANENT FILE COMMANDS 5

All time-sharing users may access permanent files after the log-in procedure is complete. Validated users may also create permanent files (refer to LIMITS Command at the end of section 4 for information concerning user validation limits).

The user number supplied during log-in represents a specific catalog in the permanent file system. The catalog contains a list of files saved under that user number as well as pertinent information about each file. All permanent file requests are made to this catalog unless an alternate user number or an auxiliary device is specified in the file request.† If an alternate user number is specified, the request is made to that user's catalog. If a validated user specifies that the file resides on an auxiliary device, the request is made to a separate catalog on that device. A separate catalog exists on the auxiliary device for each user with files on that device.

User numbers that contain asterisks (*) represent users with automatic read-only permission to files in catalogs of other users. The user number must match the alternate user number in all characters that are not asterisks. For example, a user with the user number *AB*DE* can access all files in the catalogs of the following users.

UABCDEF

UABDDEE

MABCDE1

MAB1DE3

All permanent files are classified according to the manner in which they are accessed. There are indirect access and direct access permanent files. The mode of access is determined by the command used to create the file. Indirect access permanent files are created with the SAVE command while the DEFINE command must be used to create a direct access permanent file. The command used to create a permanent file also determines the command that must be used to access the file. The user must enter the OLD, LIBRARY, or GET command in requests for indirect access files. The ATTACH command must be entered to request use of a direct access file. The following are characteristics of each type of file.

- Indirect access

  An indirect access file is, as the name implies, a permanent file that cannot be accessed directly by the user. Instead, when access is requested through entry of the OLD, LIBRARY, or GET command, a copy of the permanent file is created. The copy, considered a working file, is referenced instead of the permanent file in all subsequent I/O operations. This feature offers protection from endangering the integrity of the permanent file. Several copies of an indirect access file may exist in the system at the same time. Note that if the working file is altered, the user must enter the REPLACE command in order to update the permanent file. If the user is not allowed to update the permanent file, he can create a new indirect access file (to reflect his modifications) by entering the SAVE command and specifying a new file name.

†An auxiliary device is a mass storage permanent file device used to supplement the normal permanent file devices associated with the system (family devices).

Indirect access permanent files are allocated in blocks of 64 central memory words (one physical record unit). When compared to the size of blocks allocated for direct access files (refer to appendix D), this block size is relatively small. Thus, permanent files that are small in size are typically indirect access files. This is done to avoid allocation of more mass storage area than is actually required for the file.

- Direct access

  A direct access file is a permanent file which, upon request, becomes linked directly to the user's job. Unlike requests for indirect access files, a copy of the permanent file is not created for user access. Use of a direct access file is requested by entering the ATTACH command, and all subsequent I/O operations are performed on the permanent file itself. Note that although a copy of the permanent file is not created, a working file name may be specified in the request for a direct access file. This allows the user to reference the file by a name other than its permanent file name. Because data is written directly on the permanent file rather than on a working copy, care must be taken when modifying a direct access file.

  Direct access files have a write interlock feature. This means that if one user has attached the file in write mode, it cannot be attached by another user. Likewise, if a user wishes to attach the file in write mode, he must wait until all users currently accessing the file in alternate modes have released it.

  Direct access permanent files are allocated in large blocks (refer to appendix D) and are generally used as large data base files. When a user creates a direct access file, it is placed on the device with the most space available, unless a specific device type is specified in the request. If more than one device of the type specified exists, the file is placed on the device with the most space available.

In summary, the file access mode depends upon the command that was used to create the file. Direct access files are created with the DEFINE command. Indirect access files are created with the SAVE command. A permanent file cannot be requested as a direct access file one time using the ATTACH command, and as an indirect access file later using the OLD, LIBRARY, or GET command. The command that must be used to access the file depends upon the command that was originally used to create the permanent file.

PERMANENT FILE COMMAND FORMAT

The following describes the general format of a permanent file request. If special options are desired, they must follow the specified file name. The options must follow the / symbol, are order-independent, and are indicated by the specified keyword (kw).

pfcmd,lfn=pfn/kw_1=$option_1$,...,kw_n=$option_n$

pfcmd	Permanent file command name
lfn	Working file name used while accessing permanent file pfn. Allows user to reference a permanent file by a name other than its permanent file name.
pfn	Name under which the file is cataloged in the permanent file directory. If the pfn parameter is omitted, lfn is assumed to be the permanent file name.

The following are the keywords and corresponding option parameters which may be included.

Keyword	Option	Description
UN=	usernum	Alternate user number. This parameter is necessary only if the permanent file involved resides in another user's catalog. To access a file in another user's catalog, the requesting user must be granted explicit permission (refer to PERMIT command), must have automatic read-only permission (user number contains asterisks for all characters that do not match in the alternate user number), or the file must be a semiprivate or public file.
PW=	passwrd	The user has the option of specifying a one-to-seven character password for a file. This password must be specified whenever alternate users access the file.
CT=	n	Permanent file category. The file category determines which users may access a file. There are three categories of permanent files.
	P or PRIVATE	Files available for access only by the originating user or by those explicitly granted permission (refer to PERMIT command).
	S or SPRIV	Files available for access by all users knowing the file name, password, and user number. Accesses by alternate users for files of this category are recorded for the originator of the file. This includes the user number of the alternate user, the number of accesses made, and the date and time of the last access (refer to CATLIST command).
	PU or PUBLIC	Files available for access by all users knowing the file name, password, and user number. Only the total number of accesses to files of this category are recorded for the originator of the file.
M=	m	File or user permission mode
	W or WRITE	Allows the user to write, modify, append, read, execute, or purge the file (modify permission applies only to direct access files)
	M or MODIFY	Allows user to modify information within a direct access file and/or append information at the end of the file. The user may also read or execute the file. This mode applies only to direct access files.
	A or APPEND	Allows user to append information at the end (EOI) of the file
	R or READ	Allows user to read and/or execute the file

Keyword	Option	Description
	RM or READMD	Allows user to read a direct access file with the implication that another user may currently be accessing the file in MODIFY mode. The file may also be executed in this mode. This mode applies only to direct access files.
	RA or READAP	Allows user to read a direct access file with the implication that another user may currently be accessing the file in APPEND mode. The file may also be executed in this mode. This mode applies only to direct access files.
	E or EXECUTE	Allows the user to execute the file
	N or NULL	None; removes previously granted permission
R=	r	Specifies the type of device on which the permanent file resides or is to reside; r can be any of the following.

r	Device
DE	Extended core storage
DIn	844 disk storage subsystem
DP	Distributive data path
MDn	841 multiple disk drive
	n = Number of units ($1 \leq n \leq 8$)

The R keyword can be used in two ways.

1. It can be specified on the DEFINE command to indicate the type of family device on which the direct access file is to reside. The user should consult the service vendor to determine which devices are available. Appendix D contains a list of the block sizes allocated for each device type. If the file already exists on another device, R is ignored. If an illegal device is specified, the system responds

   DIRECT ACCESS DEVICE ERROR.

   If the number of units specified does not agree with the actual number for device types DI and MD, the system issues the message

   ILLEGAL DEVICE REQUEST.

Keyword	Option	Description
		2. It can be used in conjunction with the PN and NA keywords on any permanent file command (including DEFINE) to identify an auxiliary device on which the file resides or is to reside. If PN and NA are specified but R is not, the system default device type is assumed. If the device type of the auxiliary device PN conflicts with the device specified by the R parameter, the system issues the message ILLEGAL DEVICE REQUEST. For example: if auxiliary device PN has device type DI and the device type specified by R is MD, a conflict exists and the error message is issued.
S=	space	Specifies the amount of space in decimal PRUs desired when creating a new direct access permanent file (refer to DEFINE command). There are 640 decimal 6-bit characters in a PRU. KRONOS then creates the file on a device with the specified amount of space available. Unused space is not guaranteed to be available if the user attempts to expand the file at a later date. If no device currently has the specified amount of space available, the system responds REQUESTED SPACE UNAVAILABLE. If S is used in conjunction with the R keyword, only devices of the type specified are checked for adequate space.
PN =	packname	A one-to-seven character packname used in conjunction with the R keyword to identify an auxiliary device to be accessed in the permanent file request. This parameter is specified only when the file to be accessed resides on an auxiliary device. If the device is currently not available and the NA keyword is not specified in the request, the system responds DEVICE UNAVAILABLE. An auxiliary device is not necessarily a disk pack that can be physically removed as the parameter name implies. Rather, an auxiliary device can be any mass storage device supported by KRONOS and defined as such by the service vendor. Auxiliary devices are used to supplement the normal mass storage permanent file device associated with the system (family devices). If two or more auxiliary devices are to be in use concurrently, the user must enter the BATCH subsystem and issue the RESOURC control card image (refer to the KRONOS Reference Manual for information concerning the RESOURC control card). Note that the user must be validated to create or replace files on an auxiliary device. †

† Refer to LIMITS Command at the end of section 4 for information concerning validation limits.

Keyword	Option	Description
NA		The NA keyword specifies that if a requested resource is not available, the command will not be terminated and KRONOS will automatically suspend the user's job until the resource becomes available. NA can be used in two ways. 1. It can be specified on the ATTACH command to indicate that the user wishes to wait for the direct access file to become available. If the file is currently being accessed in a mode that conflicts with that specified in the command (interlocked), the user's job is suspended and subsequent terminal entry is not accepted. To terminate the request, enter the STOP command. 2. It can be used in conjunction with the PN and R keywords on any permanent file request (including ATTACH) for a file that resides on an auxiliary device. If the device is not currently available, the NA keyword directs the system to make the desired auxiliary device available. Note that the system will not make the device available unless the user is validated to have at least one auxiliary device assigned to his job. †
ND		The use of ND after an OLD, NEW, or LIBRARY command prevents the local files from being dropped.

PERMANENT FILE PROCESSING COMMANDS

The following list of permanent file commands is available to the user. The parameters specified have been described on the preceding pages.

APPEND,pfn,lfn_1,...,lfn_n/UN=usernum,PW=passwrd,PN=packname,R=r,NA	Appends working files lfn_1,...,lfn_n at the end of the specified indirect access permanent file pfn. The logical structure of the files is retained, that is, EORs and EOFs are appended as well as data. Each working file is appended in the order specified in the command. This command can only be used for indirect access permanent files.
ATTACH,lfn=pfn/UN=usernum,PW=passwrd,M=m,PN=packname,R=r,NA	Establishes a link to permanent file pfn for direct access usage. If pfn is omitted, the system assumes lfn=pfn.

†Refer to LIMITS Command at the end of section 4 for information concerning validation limits.

Note that a working file is not created since user access is made directly to the permanent file. Thus, the lfn parameter is used when it is desirable to reference the attached file by a name other than its permanent file name (pfn). If lfn specifies the name of a current working file, the contents of that file is lost when the permanent file is attached. In addition, lfn cannot specify the primary file name. A direct access file cannot become the primary file.

If the M keyword (permission mode) is omitted, the system assumes read permission for the attached file. The M keyword must be specified by all users, including the originator, if the file is to be modified or new information is to be added to the file.

If the NA keyword is not specified and another user is currently accessing the file in a mode that conflicts with that specified in the command (M keyword), the system responds

lfn BUSY.

If a direct access permanent file has been attached in the WRITE mode, it should be released after use in order to be attached by other users. † In general, it is best to access the file in a mode that allows other users to access the file at the same time.

A direct access permanent file can be released with any of the following commands (refer to section 4 for a description of these commands).

- OLD, NEW, or LIBRARY (all working files also released)
- GOODBYE, BYE, HELLO, or LOGIN (all working files also released)
- CLEAR (all working files also released)
- RETURN,lfn (file specified is released)

A direct access file will also be released if the lfn parameter on subsequent GET or ATTACH commands specifies the name of the attached file.

CHANGE,nfn=ofn/PW=passwrd, CT=n,M=m,PN=packname,R=r, NA

Allows the originator of a direct or indirect access permanent file to alter any of several parameters without having to attach and redefine the file or retrieve and save it. Note that this command is valid only for the originator of the file (UN keyword is illegal).

† When a direct access permanent file is attached in the WRITE mode, the last modification date is incremented even if the file is not altered.

The nfn parameter specifies the new permanent file name to be assigned; ofn is the current file name. If no name change is desired, only ofn is specified. To clear the password currently assigned to a file, set the new password equal to zero (PW=0). The CT, M, and PW keywords should be specified only if a change in the value associated with that keyword is desired. The PN and R keywords cannot be used to specify a new auxiliary device. They are used only to specify the device on which ofn resides.

If the specified permanent file, ofn, is not in the user's catalog, the system responds

ofn NOT FOUND.

If a new permanent file name, specified by nfn, already exists in the user's catalog, the system responds

nfn ALREADY PERMANENT.

The CHANGE command also updates the last modification date and last access date for the file.

DEFINE, lfn=pfn/PW=passwrd, CT=n, M=m, R=r, S=space, PN=packname, NA

Allows a validated user to create a direct access permanent file (pfn) and attach it in WRITE mode. If pfn is omitted, the system assumes lfn=pfn.

If pfn is specified, the lfn parameter can be used in two ways.

- To reference the direct access file by a name other than its permanent file name (pfn). In this case, the direct access file created contains no information initially. Data is placed on the file in succeeding write operations.
- To define a current working file as a direct access permanent file. It is important to note that the working file specified must reside on a mass storage device that also supports the residence of direct access permanent files. If the file specified by lfn does not reside on such a device, the system responds

DIRECT ACCESS DEVICE ERROR.

NOTE

Since time-sharing users are generally unaware of the device that contains their working files, this use of the lfn parameter is not recommended.

If the optional keywords are omitted, the system assumes the following values.

PW	None
CT	PRIVATE
M	WRITE
R, S	File is placed on device with most space available
PN	None
NA	Not applicable unless PN specified

If the user releases the file and then wishes to access it at some time in the future, the ATTACH command must be entered (refer to the ATTACH command).

For information regarding creation of indirect access permanent files, refer to the SAVE command later in this section.

GET, lfn=pfn/UN=usernum, PW=passwrd, PN=packname, R=r, NA

Retrieves a copy of the specified indirect access permanent file (pfn) for use as a working file. This command can only be used for indirect access permanent files. If pfn is omitted, the system assumes lfn=pfn.

If the user wishes to reference the working file by a name other than its permanent file name (pfn), the lfn parameter is specified. However, if lfn specifies the name of a current working file, the contents of that file will be replaced by the copy of pfn. The current primary file and subsystem in use remain the same unless the file name specified for lfn is that of the current primary file. In this case, the contents of the primary file is replaced by the copy of pfn which becomes the new primary file. Furthermore, if pfn has a subsystem flag associated with it (refer to SAVE command), that subsystem is selected automatically. For example, if file A is the current primary file and the user enters

```
GET, A=B
```

the copy of permanent file B will become the new primary file. If a subsystem other than the one currently active is associated with file B, it is also selected.

If the request is for a file in another user's catalog (UN keyword specified), the permission mode (READ, WRITE, EXECUTE, etc.) is that which the user has been permitted for private files or that specified in the catalog for semi-private and public files.

LIBRARY, lfn=pfn/PW=
passwrd, PN=packname

Retrieves a copy of the specified permanent file from the catalog of special user number LIBRARY. This command can only be used for indirect access permanent files.

The permission mode is that which has been granted for private files or specified in the catalog for semiprivate and public files. The copy of the permanent file retrieved becomes the new primary file and all other working files are released unless the next command entered is NODROP.

The ATTACH, GET, or OLD command can also be used to access permanent files in the catalog of user number LIBRARY. For additional information, refer to LIBRARY Command at the end of this section.

OLD, lfn=pfn/UN=usernum,
PW=passwrd, PN=packname, ND

Retrieves a copy of the specified permanent file (indirect access only) for use as the primary file. If a specific subsystem is associated with the file, it is selected automatically. This occurs only if the file was originally the primary file and was saved while a subsystem other than NULL was active. Refer to the description of the SAVE command in this section for additional information. All working files are released unless the ND parameter is specified.

PACKNAM, PN=packname
or
PACKNAM, packname

Enables validated users to direct subsequent permanent file requests to the auxiliary device specified by the packname parameter. This command allows the user to omit the PN keyword in requests for files that reside on that auxiliary device. However, if permanent files on another auxiliary device are to be accessed, the PN keyword can be included in the request or a new PACKNAM command can be entered.

The user cannot access permanent files residing on the normal system devices (family devices) while the PACKNAM command is in effect. To access these files again, he must enter the PACKNAM command and set the packname parameter to zero, as follows:

PACKNAM, PN=0 or PACKNAM

An auxiliary device is not necessarily a disk pack that can be physically removed as the command name implies. Rather, an auxiliary device can be any mass storage device supported by KRONOS and defined as such by the service vendor. Auxiliary devices are used to supplement the normal mass storage permanent file devices associated with the system (family devices).

Examples:

```
OLD,FILEA  <------ Request to normal family
                   device for file FILEA

PACKNAM,   <------ Permanent file requests
 PN=ABC            that follow are made to
      .            auxiliary device ABC,
      .            unless PN option is speci-
      .            fied in the request or a
      .            new PACKNAM command
                   is entered
      .
GET,DATA   <------ Request to auxiliary device
      .            ABC for file DATA
      .
PACKNAM,   <------ Permanent file requests
 PN=XYZ            that follow are made to
      .            auxiliary device XYZ
      .
GET,TAPE1  <------ Request to auxiliary device
GET,TAPE2          XYZ for files TAPE1 and
      .            TAPE2
      .
PACKNAM,   <------ Clear PACKNAM command.
 PN=0              Permanent file requests
      .            that follow are made to the
      .            normal family devices
      .
```

PERMIT,pfn,usernum$_1$=m$_1$,
usernum$_2$=m$_2$,...,usernum$_n$=m$_n$/
PN=packname,R=r,NA

Grants user usernum$_i$ permission to access private file pfn. If the permission mode, m$_i$, is omitted, the system assumes read permission.

PURGE,pfn/UN=usernum,
PW=passwrd,PN=packname,
R=r,NA

Removes the specified permanent file from permanent file storage. If the permanent file does not exist, the system responds

pfn NOT FOUND.

WRITE permission is required to purge a permanent file in an alternate user's catalog (UN keyword specified). The PW keyword must also be specified if the file has a password. It is important to note that direct access files are not actually purged until the last user accessing the file has released it. However, subsequent attempts to access the file will be rejected.

REPLACE,lfn=pfn/UN=usernum,
PW=passwrd,PN=packname,
R=r,NA

Allows validated users to replace the contents of a permanent file (pfn) with the contents of a working file (lfn). If the file name specified for pfn does not exist, a new permanent file is created. This command can only be used for indirect access permanent files. Note that the REPLACE command does not affect the subsystem associated with the permanent file (if any).

SAVE, lfn=pfn/PW=passwrd,
CT=n, M=m, PN=packname, R=r, NA

Creates an indirect access permanent file. This command allows a validated user to retain a copy of the specified working file in the permanent file system. If the optional keywords are omitted, the system assumes the following values.

PW	None
CT	PRIVATE
M	WRITE
PN	None
R, NA	Not applicable unless PN is specified

Unless the NULL subsystem is active when the primary file is saved, an internal indicator called the subsystem flag is set to indicate which subsystem is being used. That subsystem becomes associated with the permanent file and is selected automatically each time the file is retrieved in subsequent requests using the OLD command. To save the primary file without a subsystem association, the user must enter the NULL subsystem before issuing the SAVE command. Note that the subsystem flag is set only when saving the primary file. There is no subsystem association when working files other than the primary file are saved.

CATLIST COMMAND

The user may wish to obtain specific information about his permanent files. This can be done with the CATLIST command. In addition, this command may be entered if the user wishes to obtain specific information about permanent files that he can access in catalogs of alternate users. The following is the format of the CATLIST command; the optional keywords (and corresponding parameters) to be included must follow the / symbol and are order-independent.

CATLIST/LO=options, FN=pfn, UN=usernum, L=lfn, PN=packname, R=r, NA

Keyword	Option	Description
LO=	F	Selects a listing of pertinent information about each file in the user's catalog (refer to example 1). If an alternate user number is specified (UN keyword), the user obtains a listing of all files that he can access in the alternate user's catalog. Note that the password for files in an alternate user's catalog is not included in the listing. The password to files in an alternate user's catalog must be obtained directly from that user (refer to example 2).

Keyword	Option	Description
	FP	Selects a listing of permission information recorded for each alternate user that has accessed a specified file in the user's catalog (refer to example 3). This option requires that a file name be specified (FN keyword). If an alternate user number is also specified (UN keyword), only the permission information recorded for that user of the specified file is listed. The user numbers listed include: • Those that have been granted explicit permission to the file (private file only). • Those that have accessed the file because of implicit permission (semiprivate files only). Permission information for alternate users that have accessed public files is not recorded.
	0 (zero)	Selects a short list that indicates only the names of the files in the user's catalog (refer to example 4). If an alternate user number is specified (UN keyword), the user obtains only the names of the files that he can access in the alternate user's catalog. If no LO keyword is specified, the system assumes this value.
	P	Selects a short list that indicates only the user numbers of alternate users who have accessed the specified private or semiprivate file (refer to example 5). This option requires that a file name be specified (FN keyword).
FN=	pfn	Permanent file name. This option specifies that catalog information is desired only for this permanent file. This parameter is required when listing permit information (LO=FP, LO=P). If the short list options are selected (LO=0, LO=P), the message pfn FOUND. is returned if the file (or user number) is located. The message pfn NOT FOUND. is returned if specified file (or user number) is not located.
UN=	usernum	User number. This parameter has two purposes. For LO=F and LO=0. Indicates the alternate catalog for which the user desires catalog information (refer to example 2). For LO=FP and LO=P. Indicates the permission information recorded for the specified alternate user.

Keyword	Option	Description
L=	lfn	Output file name. This is the name of a working file to which the CATLIST information is written. If this parameter is omitted, the system assumes L=OUTPUT and the data is returned to the terminal. If lfn is a current working file, the CATLIST information is written at the current position in the file. For example, if lfn is positioned at the beginning-of-information (BOI), the contents of that file is replaced with the CATLIST information. If lfn is positioned at end-of-information (EOI), the CATLIST information is appended to the file as a new logical record.
R=	r	This keyword is used in conjunction with the PN and NA keywords to specify the type of auxiliary device that contains the desired catalog information. If PN and NA are specified but R is not, the system default device type is assumed. If the device type of the auxiliary device PN conflicts with the device specified by the R parameter, the system issues the message ILLEGAL DEVICE REQUEST. For example: if auxiliary device PN has device type DI and the device type specified by R is MD, a conflict exists and the error message is issued. The device types are: r — Device DE — Extended core storage DIn — 844 disk storage subsystem DP — Distributive data path MDn — 841 multiple disk drive n = number of units ($1 \leq n \leq 8$) If the number of units specified does not agree with the actual number for device types DI and MD, the system issues the message ILLEGAL DEVICE REQUEST.

Keyword	Option	Description
PN=	packname	The packname option identifies an auxiliary device that contains catalog information for all users with files on that device. Unless the PACKNAM command is in effect, the PN keyword must be specified if a user wishes to obtain the following information from his catalog on an auxiliary device. • Pertinent information about each file that resides on the device (LO=F) • Only the name of each file (LO=0) • Permission information for each alternate user that has accessed a specific file (LO=FP) • Only the user number of each alternate user that has accessed a specific file (LO=P) If two or more auxiliary devices are to be in use concurrently, the user must enter the BATCH subsystem and issue the RESOURC control card image (refer to the KRONOS Reference Manual for information concerning the RESOURC control card). If an auxiliary device is available in the system, the PN keyword can also be specified by alternate users to obtain a list of files and pertinent information about each file they can access on the device.
NA		This keyword is used in conjunction with the PN keyword in requests for catalog information on auxiliary devices. If the device is currently not available, NA directs the system to make the desired device available. If the NA keyword is omitted and the auxiliary device is not currently available, the system responds DEVICE UNAVAILABLE. Note that the system does not make the desired device available unless the user is validated to have at least one auxiliary device assigned to his job (refer to LIMITS command in section 4).

CATLIST EXAMPLES

The following examples illustrate some of the options available to the user when entering the CATLIST command (the user entering the command in the examples is USER123).

Example 1. Listing of pertinent information about each file in the catalog of USER123. The command is entered in the form ... CATLIST/LO=F

CATALOG OF USER123 75/04/10. 13.41.59.

	FILE NAME PASSWORD	ACCESS MD/CNT	FILE-TYPE		LENGTH PERM.	DN SUBSYS	CREATION DATE/TIME	LAST ACCESS DATE/TIME	LAST MOD DATE/TIME
1	DIRFILE CDC	DIR. 13	PUBLIC	READAP	25600	40	75/01/13. 19.02.50.	75/03/23. 07.33.05.	75/03/23. 07.33.05.
2	DIRTEST CDC	DIR. 4	PUBLIC	MODIFY	17280	40	75/01/13. 18.59.05.	75/01/13. 19.49.40.	75/01/13. 18.59.05.
3	ABC 1234567	IND. 24	PRIVATE	WRITE	1280		75/01/12. 06.42.13.	75/03/23. 07.30.35.	75/01/13. 21.16.52.
4	TAPE1	IND. 20	SEMI-PR	READ	640 FTNTS		75/01/13. 19.58.53.	75/03/23. 07.30.53.	75/01/13. 21.17.54.
5	PRIME ABCDEFG	IND. 22	SEMI-PR	READ	1280 BASIC		75/01/12. 05.38.37.	75/03/23. 07.32.19.	75/02/09. 13.55.49.
6	TAPE2	IND. 12	SEMI-PR	APPEND	1920 BASIC		75/01/12. 06.27.00.	75/03/23. 07.31.27.	75/01/13. 21.20.03.
7	BINTAPE BINARY	IND. 8	PUBLIC	EXEC	11520 EXEC.		75/01/12. 06.37.03.	75/01/13. 21.42.54.	75/01/13. 21.42.54.

Subsystem flag associated with file when it was saved. Blank entry indicates that the NULL subsystem was being used or the file was not the primary file when the SAVE command was issued.

Example 2. Listing of pertinent information about each file that USER123 can access in the catalog of USER456. The command is entered in the form... CATLIST/LO=F,UN=USER456

CATALOG OF USER123 75/04/10. 13.44.53.

ALTERNATE CATALOG USER456

	FILE NAME	ACCESS	FILE-TYPE	LENGTH	DN	CREATION	LAST ACCESS	LAST MOD
		MD/CNT	PERM.	SUBSYS		DATE/TIME	DATE/TIME	DATE/TIME
1	SAMPLE	IND. 7	PUBLIC	640		75/01/13. 19.28.29.	75/01/13. 20.13.30.	75/01/13. 19.28.29.
			READ EXEC.					
2	SBMTJOB	IND. 7	PUBLIC	1920		75/01/13. 19.22.44.	75/02/09. 14.18.51.	75/01/13. 19.24.21.
			READ BASIC					
3	EXAM	IND. 14	SEMI-PR	2560		75/01/13. 19.29.58.	75/02/09. 14.12.08.	75/02/09. 14.02.58.
			WRITE FTNTS					
4	LOG10	DIR. 6	SEMI-PR	25600	*	75/01/13. 19.14.06.	75/02/09. 14.12.44.	75/01/13. 20.29.14.
			MODIFY					
5	DEMO	DIR. 4	PUBLIC	17280	*	75/01/13. 19.18.52.	75/02/09. 14.02.05.	75/02/09. 14.02.05.
			READMD					

DN indicates device number for direct access files. * in this column indicates file resides on master device (refer to glossary for definition).

Example 3. Listing of permission information recorded for each user that has accessed file PRIME (semiprivate file) in the catalog of USER123. The command is entered in the form... CATLIST/LO=FP,FN=PRIME

CATALOG OF USER123 75/04/10. 13.47.50.

FILE NAME PRIME

USER NUMBER		PERM.	ACCESSES	DATE	TIME
1.	USER456	READMD	1	75/01/13.	20.18.09.
2.	JIMALTE	READMD	2	75/01/13.	20.44.27.
3.	DICKOSW	READMD	1	75/01/13.	20.45.28.
4.	EARLDRE	READMD	3	75/01/13.	20.48.20.
5.	CRAIGGE	READMD	1	75/01/13.	20.49.39.
6.	SONYAAN	READMD	2	75/01/13.	20.51.37.

Example 4. List of current files in the catalog of USER123. The command is entered in the form ... CATLIST/LO=0 (it is not necessary to specify the LO=0 option since it is the default value).

```
CATALOG OF USER123                    75/04/10. 13.49.18.

  FILE NAME(S)

DIRFILE   DIRTEST   ABC       TAPE1   PRIME   TAPE2   BINTAPE
          7 FILE(S)
```

Example 5. List of alternate users that have accessed file DIRFILE in the catalog of USER123. The command is entered in the form... CATLIST/LO=P, FN=DIRFILE

```
CATALOG OF USER123                    75/04/10. 13.51.28.

              FILE NAME DIRFILE
  USER NUMBER(S)

USER456     JIMALTE     DICKOSW

            3 USER(S)
```

ALTERNATE USER ACCESS

The following examples illustrate the alternatives available for users to access files in alternate catalogs.

PRIVATE FILE CATEGORY

Working file ABC is retained by USER123 as a private indirect access permanent file using the following command.

```
SAVE,ABC/PW=1234567
```

The following is the entry for file ABC in USER123's catalog. The listing is obtained by entering the CATLIST command in the form ... CATLIST/LO=F,FN=ABC

```
CATALOG OF USER123                                    75/04/10. 13.53.02.

    FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION LAST ACCESS LAST MOD
      PASSWORD MD/CNT          PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME

1   ABC       IND.   PRIVATE      1280    75/01/12.  75/03/23.  75/01/13.
    1234567     24          WRITE         06.42.13.  07.30.35   21.16.52.
```

User USER123 issues the following command to permit alternate user access.

PERMIT,ABC,USER456=R,JIMALTE=W

The following is the permission information recorded in the catalog of USER123. The listing is obtained by entering the CATLIST command in the form ... CATLIST/LO=FP,FN=ABC

```
CATALOG OF USER123                                    75/04/10. 13.54.18.

                FILE NAME ABC
USER NUMBER     PERM.    ACCESSES  DATE      TIME

 1. USER456      READ           1  75/01/13. 21.29.29.
 2. JIMALTE     WRITE           2  75/01/13. 21.32.33.
```

The format of the commands required by USER456 and JIMALTE to access the file is:

GET,ABC/UN=USER123,PW=1234567 or OLD,ABC/UN=USER123,PW=1234567

Only the following users may access this private file.

User	Information Necessary For Access	
USER123	File name	ABC
USER456	File name	ABC
	File password	1234567
	User number	USER123
JIMALTE	File name	ABC
	File password	1234567
	User number	USER123

SEMIPRIVATE FILE CATEGORY

Working file TAPE1 is retained by USER123 as a semiprivate indirect access permanent file using the following command.

```
SAVE,TAPE1/CT=SPRIV,M=READ
```

The following is the entry for file TAPE1 in USER123's catalog. The listing is obtained by entering the CATLIST command in the form... CATLIST/LO=F,FN=TAPE1

```
CATALOG OF USER123                    75/04/10. 13.55.19.

FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION LAST ACCESS LAST MOD
  PASSWORD MD/CNT          PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME

1  TAPE1     IND.  SEMI-PR        640     75/01/13.  75/03/23.  75/01/13.
               20          READ  FTNTS    19.58.53.  07.30.53.  21.17.54.
```

The following is the permission information recorded in the catalog of USER123. The listing is obtained by entering the CATLIST command in the form... CATLIST/LO=FP, FN=TAPE1

```
CATALOG OF USER123                    75/04/10. 13.56.18.

                FILE NAME TAPE1
USER NUMBER     PERM.   ACCESSES    DATE      TIME

1.  JIMALTE     READ        2    75/01/13.  21.47.34.
2.  USER456     READ        3    75/01/13.  21.48.43.
3.  DICKOSW     READ        1    75/01/13.  21.49.53.
4.  SONYAAN     READ        4    75/01/13.  21.51.09.
5.  EARLDRE     READ        2    75.01/13.  21.53.00.
6.  CRAIGGE     READ        1    75/01/13.  21.54.03.
```

This indicates that six users have accessed file TAPE1 the specified number of times. Whenever an alternate user accesses file TAPE1 the first time, the permission information is recorded in the catalog of USER123. Each succeeding access increments the access count (ACCESSES).

The format of the commands required by all users, except USER123, to access the file is:

GET,TAPE1/UN=USER123 or OLD,TAPE1/UN=USER123

Users who may access this file are:

User	Information Necessary For Access	
USER123	File name	TAPE1
All others	File name	TAPE1
	Password	None required
	User number	USER123

PUBLIC FILE CATEGORY

Working file BINTAPE is retained by USER123 as a public indirect access permanent file using the following command.†

SAVE,BINTAPE/PW=BINARY,CT=PUBLIC,M=EXECUTE

The following is the entry for file BINTAPE in USER123's catalog. The list is obtained by entering the CATLIST command in the form... CATLIST/LO=F,FN=BINTAPE

CATALOG OF USER123 75/04/10. 13.57.40.

	FILE NAME	ACCESS PASSWORD	FILE-TYPE MD/CNT	LENGTH	DN PERM.	CREATION SUBSYS	LAST ACCESS DATE/TIME	LAST MOD DATE/TIME
1	BINTAPE	IND.	PUBLIC	11520		75/01/12.	75/01/13.	75/01/13.
	BINARY	8		EXEC	EXEC.	06.37.03.	21.42.54.	21.42.54.

Permission information is not recorded for accesses to a public file. Only the total number of accesses to files of this category are recorded for the originator of the file.

The format of the command required by all users, except USER123, to access file BINTAPE is:

GET,BINTAPE/UN=USER123,PW=BINARY
or
OLD,BINTAPE/UN=USER123,PW=BINARY

Users who may access this file are:

User	Information Necessary For Access	
USER123	File name	BINTAPE
All others	File name	BINTAPE
	Password	BINARY
	User number	USER123

† The file category LIBRARY which existed in previous versions of KRONOS remains valid for upward compatibility. This category was renamed PUBLIC to avoid confusion with the permanent file command LIBRARY.

LIBRARY COMMAND

The LIBRARY command allows access to indirect access permanent files in the catalog of special user number LIBRARY. Direct access files may also reside in the catalog of user number LIBRARY although an alternate method must be employed to access these files (described later in this section). The file category may be private, semiprivate, or public. The permission mode is that which has been granted for private files or specified in the catalog for semiprivate and public files.

When the LIBRARY command is entered, a copy of the specified indirect access permanent file is retrieved. This copy becomes the new primary file and all other working files are dropped unless the ND parameter is used. For example:

```
LIBRARY,ABC75/PW=13479,ND
```

In this example, a copy of the indirect access permanent file ABC75 is retrieved and becomes the new primary file. Note that only indirect access permanent files may be accessed with the LIBRARY command. The following commands may also be entered to access permanent files in the catalog of user number LIBRARY (differences in command function are described).

- ATTACH,lfn=pfn/UN=LIBRARY,PW=passwrd,M=m

  This command must be entered in order to use any direct access permanent files in the catalog of user number LIBRARY. Note that a working file is not created since all I/O operations are performed directly on the permanent file itself. However, if the user wishes to reference the file by a name other than its permanent file name, he may assign a working file name (lfn) in the command. The primary file name remains unchanged.

- GET,lfn=pfn/UN=LIBRARY,PW=passwrd

  This command retrieves a copy of the specified indirect access permanent file (from catalog of LIBRARY) for use as a working file. The primary file remains unchanged unless lfn specifies the name of the current primary file (refer to description of GET command for additional information).

- OLD,lfn=pfn/UN=LIBRARY,PW=passwrd

  The function of this command is identical in all respects to that of the LIBRARY command.

The only files that can be accessed from the catalog of LIBRARY are those that allow alternate user access. Note that all users, except user number LIBRARY, are considered to be alternate users. The rules for alternate user access are summarized as follows:

- Only those users permitted to private files may access them.

- All users can access semiprivate and public files providing they know the file name, password (if one exists for the file), type of access (direct or indirect), and user number (LIBRARY).

The following forms of the CATLIST command may be entered to obtain a listing of files that can be accessed in the catalog of user number LIBRARY.

```
CATLIST/LO=F,UN=LIBRARY (full listing)
CATLIST/UN=LIBRARY (file names only)
```

TIME-SHARING JOB PROCESSING 6

This section describes various conventions used in time-sharing job processing, as well as facets of system operation.

INPUT/OUTPUT CONVENTIONS

The following conventions and standards are used for I/O operations at a time-sharing terminal.

- ASCII/NORMAL modes

  Under NORMAL mode, a 64-character ASCII subset (possibly 63 characters depending upon installation option) is used. This character set is identical to the BATCH character set, allowing the time-sharing user full compatibility with batch.

  Under ASCII mode, a 96-character ASCII subset is used. Under this mode, the 74 and 76 codes are interpreted as escape codes and not as data. Refer to appendix C for details of control bytes for I/O.

- Standard input lines

  The standard input line entered from the terminal keyboard consists of commands or information (program statements, data, or text). Each input line containing information is placed in the primary file and must be numbered. A space must ordinarily be inserted between the line number and the coding. An exception is the FORTRAN Extended continuation line which has a + immediately after the line number. Line numbers for FORTRAN Extended or BASIC programs cannot be more than five digits. All other coding can have line numbers up to ten digits. Refer to the description of file sorting in section 3 for additional information.

- Text mode input

  Text mode allows keyboard entry of information into the primary file without specifying line numbers. Lines entered in text mode are interrupted as text and are not processed by the system as commands. For complete information concerning text mode, refer to the TEXT command described in section 4.

- Text Editor input

  Normally, keyboard entry of information is possible only to the primary file. However, the Text Editor allows the user to enter or modify information in all working files (including the primary file) from the terminal keyboard.

  An important feature of the Text Editor is its ability to reference direct access permanent files. Since a direct access file cannot be referenced with the OLD command, it cannot become the primary file, and normal keyboard entry to the file is therefore impossible. The Text Editor is one method that is commonly used to accomplish keyboard entry of source code into a direct access file. The following example illustrates the use of the Text Editor to enter source code into an empty direct access permanent file from the terminal keyboard (note that a direct access file is empty when it is created).

```
DEFINE, DIRFILE
READY.
EDIT, DIRFILE
BEGIN TEXT EDITING
? ADDS
ENTER TEXT.
? /THIS IS A TEST EXAMPLE TO ILLUSTRATE THE ABILITY TO ENTER
? TEXT INTO A DIRECT ACCESS PERMANENT FILE DIRECTLY FROM THE
? TERMINAL KEYBOARD.  DATA OR NUMBERED PROGRAM STATEMENTS CAN
? ALSO BE ENTERED IN THE SAME MANNER./
READY.
? END
END TEXT EDITING
SRU          0.019 UNTS.
CP           0.019 SECS.
READY.

LNH, F=DIRFILE
THIS IS A TEST EXAMPLE TO ILLUSTRATE THE ABILITY TO ENTER
TEXT INTO A DIRECT ACCESS PERMANENT FILE DIRECTLY FROM THE
TERMINAL KEYBOARD.  DATA OR NUMBERED PROGRAM STATEMENTS CAN
ALSO BE ENTERED IN THE SAME MANNER.

SRU          0.007 UNTS.
READY.

RETURN, DIRFILE

READY.
```

.
.
.

For complete information concerning the use of the Text Editor, refer to the Text Editor Reference Manual. A list of the commands available under the Text Editor is included as appendix E of this manual.

Although data can be entered into a direct access file from the terminal keyboard, through use of the Text Editor, a program can also be used. Since direct access files are normally large in size, a program is generally used to manipulate or maintain data on direct access files.

- Length of input/output lines

  The input line may consist of 150 characters maximum. The output line may be any length in accordance with the line width of the terminal being used. If the user attempts to input or output more characters per line than are allowed, the additional characters overprint at the end of the line, unless the terminal being used has an automatic carriage return feature.

- Terminate the input line

  The user must terminate each line of input information by pressing the RETURN key. This tells the system that the current input line is complete. KRONOS responds by positioning the carriage to the beginning of the next line. The user can then enter additional input information on the new line.

- Correct the input line

  Entry errors in the input line can be corrected, before the RETURN key is pressed, through use of the backspace character. The backspace character, on ASCII code terminals, is the underline (_) key.† On correspondence code

† The backspace character on Teletype model 33/35 terminals is the shift-Ø key. The character is an underline (_).

terminals, the BACK SPACE key is used. One character (including spaces) is deleted for each backspace character entered. If the beginning of the line is reached, further backspace characters are ignored. For example, if the input line contained ...

BAX_SJK_ _IC

it would be interpreted by KRONOS as the subsystem selection command BASIC.

- Delete the input line

  If the user discovers an error in the command or other input just typed, he can delete the current input line by pressing the ESC or ESCAPE key (ASCII code terminals) or the ATTN key (correspondence code terminals). This must be done before the RETURN key is pressed. KRONOS ignores the entire input line and responds by printing *DEL* and positioning the carriage to the beginning of the next line. The following example illustrates the use of the ESC key.

```
NEW,TEST
READY.

10 PROGRAM T(OUTPUT)
20 PRINT 6
30 6 FORMAT (*THIS IS A TEST*9 _ )  *DEL*  <-- ESC key pressed
40 6 FORMAT (*THIS IS IT*)
50 END
LNH

10 PROGRAM T(OUTPUT)
20 PRINT 6
40 6 FORMAT (*THIS IS IT*)
50 END
READY.
```

- Input to executing program

  A ? output to the terminal normally indicates that the executing program has requested input. However, it should be noted that the program may include question marks in its normal output.

- Interrupt executing program

  To interrupt an executing program that is currently transmitting output to an ASCII code terminal, press the I key† or the interrupt key (generally labeled INT, INTRPT, or BREAK on most ASCII code terminals). If a correspondence code terminal is being used, press the ATTN key.

  This process is commonly referred to as job suspension and is described in more detail in section 7.

- Terminate executing program

  To terminate an executing program that is currently transmitting output to an ASCII code terminal, press the S key. If a correspondence code terminal is being used, press the ATTN key. If the program is in execution, but is not actively transmitting output, enter the STOP command to terminate the program.

---

†If the program in execution is not currently transmitting output to the terminal, the I key cannot be used to interrupt the program. However, the other keys described can still be used.

- System messages during input

  The following diagnostic messages can appear at anytime during input from the terminal.

OVL	Line overflow. This message is issued when more than 150 characters have been entered since the last carriage return (the entire line is lost when the 151st character is entered).
RE-ENTER LAST LINE	Data was lost during the last line of input. The user should reenter the last line.

TERMINAL JOB PROCESSING

This section explains the basic steps in processing a time-sharing job. These steps include:

- Terminal log-in
- Subsystem selection
- Program entry and execution
- Terminal log-off

TERMINAL LOG-IN

Refer to section 2 for the full log-in procedure.

SUBSYSTEM SELECTION

1. Perform step a or b depending upon which system response was given during log-in.

   a. If the RECOVER/SYSTEM: response was issued, enter the subsystem to be used or any valid command (refer to sections 4 and 5).

BASIC (CR)	To use the BASIC language
BATCH, nnnnn (CR)	To enter control card images interactively from the terminal; nnnnn is the field length to be used
EXECUTE (CR)	To execute a previously compiled program
FTNTS (CR)	To use the time-sharing FORTRAN Extended language

To BASIC, FTNTS, or EXECUTE, the system responds:

```
OLD, NEW, OR LIB FILE:
```

To BATCH, nnnnn the system responds:

```
$RFL, nnnnn.                (nnnnn is the specified field length)
/
```

NOTE

A subsystem may be automatically selected by the system at log-in if this has been specified in the system validation file. This validation file determines the internal system controls associated with each valid user number. Refer to the discussion of the LIMITS command at the end of section 4 for further information.

b. If the RECOVER/CHARGE: response was issued, type CHARGE, followed by a valid charge number and project number.

CHARGE, chargeno, projectno (CR)

The system responds:

READY.

The user then enters the name of the subsystem to be used or any valid command (refer to sections 4 and 5).

BASIC (CR)	To use the BASIC language
BATCH, nnnnn (CR)	To enter control card images interactively from the terminal; nnnnn is the field length to be used
EXECUTE (CR)	To execute a previously compiled program
FTNTS (CR)	To use the time-sharing FORTRAN Extended language

To BASIC, FTNTS, or EXECUTE, the system responds:

OLD, NEW, OR LIB FILE:

To BATCH, nnnnn the system responds:

$RFL, nnnnn. (nnnnn is the specified field length)
/

2. Enter the appropriate file type.

OLD (CR)	For a file that was previously saved as a permanent file (the file cannot be a direct access file)
NEW (CR)	For a new working file
LIBRARY (CR)	For an indirect access permanent file that resides in the catalog of user number LIBRARY

The system responds:

FILE NAME:

For responses to BATCH requests, refer to section 9.

3. Enter the file name (one to seven characters).

lfn (CR)

If an OLD or LIBRARY file is specified, but does not exist, the system responds:

pfn NOT FOUND. (pfn is the name of the file requested)

The user must then return to step 8 and attempt to enter the file type and file name again.

If no errors are detected, the system responds:

READY.

At this point, the output at the terminal has the following form:

```
 yy/mm/dd.  hh.mm.ss
 KRONOS TIME-SHARING SYSTEM - VER. 2.1.2
USER NUMBER: usernum
PASSWORD
XXXXXXXXX  <-------------------  Password, if present, is not visible
TERMINAL: nnn, iii                through black out.
RECOVER/SYSTEM: BASIC
OLD, NEW, OR LIB FILE: NEW
FILENAME: PRIME
READY.
```

Now, if the user wishes, he can vary the source and format of information sent to or received from the system by entering any of the terminal control commands (section 4). All of these commands, except AUTO and TAPE, may be automatically selected by the system at log-in, depending on the validation parameters that have been established for the user's particular site.

To hasten the preceding log-in sequence, the user may select the short form as follows:

```
yy/mm/dd.  hh.mm.ss
KRONOS TIME-SHARING SYSTEM - VER. 2.1.2
USER NUMBER: USER123,ABCDEFG  <-------  User number and password
                                        entered on same line
TERMINAL: 40,     TTY
RECOVER/SYSTEM: BASIC,OLD,DEMO  <-------  Subsystem, file type, and
                                          file name entered on
READY.                                    same line
```

Note that any time the OLD command is used in requests for indirect access permanent files (refer to preceding examples), the subsystem need not be specified unless the file requested has a NULL subsystem flag associated with it. If the file has a subsystem flag other than NULL associated with it (that is, BASIC, FTNTS, EXECUTE, or BATCH), that subsystem is selected automatically. A subsystem flag becomes associated with a file when it is saved (refer to the SAVE command in section 5).

In the following example, the user logs into the system and requests file DEMO with the OLD command. He does not specify a subsystem since he knows that the subsystem flag associated with the file causes BASIC to be selected automatically.

```
yy/mm/dd.  hh.mm.ss
KRONOS TIME-SHARING SYSTEM - VER. 2.1.2
USER NUMBER:  USER123,ABCDEFG
TERMINAL:  40,    TTY
RECOVER/SYSTEM:  OLD,DEMO  <-------- The BASIC subsystem flag
                                     associated with file DEMO causes
READY.                               BASIC to be selected automatically.
```

To ensure that the BASIC subsystem has indeed been selected, the user can enter the ENQUIRE command. The system responds:

```
TERMINAL:  40,    TTY
SYSTEM - BASIC
FILE NAME:  DEMO
STATUS - IDLE
```

PROGRAM ENTRY AND EXECUTION

Perform all necessary file creation and/or file retrieval activity. If new BASIC or FORTRAN programs are entered, each line must be entered in the proper format and must begin with a one- to five-digit line number. Refer to the FORTRAN Extended, version 4, and/or BASIC Reference Manuals for complete information concerning each language.

Enter the following command to compile and/or execute a BASIC or FORTRAN program.

```
RUN    (CR)
```

This command causes the FORTRAN Extended or BASIC program to be compiled and executed. The RUN command is also used to execute a program under the EXECUTE subsystem.

The output of a BASIC or FORTRAN Extended program is in the form:

```
yy/mm/dd.    hh.mm.ss.
PROGRAM      nnnnnnn
               .
               .
(data requested or gener-
ated by the program;
error messages if pro-
gram errors occurred)
               .
               .
SRU 0.032 UNTS. †
RUN COMPLETE.
```

† The output of SRU units (for the preceding activity) after completion of job execution occurs only in the BASIC, FTNTS, and EXECUTE subsystems. The total accumulated SRUs under the current charge/project number may be determined in all subsystems by entering the ENQUIRE,S command.

If the user wishes merely to compile object code which can be saved and used at a later time, he enters the command

RUN, B=lfn (CR)

where lfn is the name of the file on which the object code is written.

If the user wishes to retain the compiled program for later execution under the EXECUTE subsystem, the SAVE command must be entered (no subsystem name is retained when a working file, other than the primary file, is saved).

Refer to sections 4 and 5 in this manual for complete information concerning the time-sharing commands available to the user. These commands include:

- Terminal control commands
- Subsystem selection commands
- Time-sharing job commands
- Permanent file commands

Appendix G of this manual illustrates a sample session at the terminal.

TERMINAL LOG-OFF

Refer to section 2 for logging-off.

ADDITIONAL SYSTEM FEATURES 7

This section contains a description of the program editing, job suspension, and recovery features provided for the KRONOS terminal user.

PROGRAM EDITING

To allow ease of editing the primary file when working with source programs, KRONOS provides a simplified editing capability that can be used under any subsystem.† This capability allows the user to:

- Insert statements into a program
- Delete statements from a program
- Correct statements in a program

The following example illustrates this program editing capability. The user types:

```
10 LET A=1 (CR)
20 LET B=50 (CR)
30 FOR I=1 TO B (CR)
40 A=A*I (CR)
50 PRINT "FACTORIAL" ; I,A (CR)
60 NEXT I (CR)
70 END (CR)
```

If the user wishes to insert a statement between line numbers 10 and 20, he types:

```
15 LET Z=20 (CR)
```

If the user wishes to delete the statement at line 20, he types:

```
20 (CR)
```

Finally, to change the statement at line number 30, the user types:

```
30 FOR I=1 TO Z (CR)
```

Leading zeros are not required even if the statement being deleted or corrected contains line numbers with leading zeros. For example, a line numbered 30 would replace and be interchangeable with a line numbered 00030.

† For more detailed editing, the Text Editor should be used. Refer to the Text Editor (EDIT) Reference Manual for complete information concerning its use.

When the file is listed, it has the following form.

```
10 LET A=1
15 LET Z=20
30 FOR I=1 TO Z
40 A=A*I
50 PRINT "FACTORIAL" ; I,A
60 NEXT I
70 END
```

To replace the new version of the file in the permanent file system, use:

```
REPLACE
```

If the user wishes to save the file under a new name, he may enter:

```
SAVE,    lfn=pfn
```

lfn Primary working file name
pfn Permanent file name

RESEQUENCING LINE NUMBERS

It is often convenient to resequence the line numbers of the primary file. This is done using the following command.

```
RESEQ,nnnnn,iiii,t    (CR)
```

Where:

nnnnn — New line number of the first statement (five digits maximum); if omitted, the system assumes nnnnn=00100

iiii — Increment to be added to nnnnn; if omitted, the system assumes iiii=0010

t — Type of file to be resequenced

- B — For files that contain BASIC source code. BASIC statements that contain references to line numbers are updated when the line numbers are resequenced.
- T — For files that contain text source information. A five-digit line number, plus a blank, is added at the beginning of each line. No inspection is made for existing line numbers. Thus, if line numbers currently exist, they are not resequenced or deleted and two sets of line numbers result.
- other — Any number at the beginning of a line is considered a line number and is resequenced according to the nn and ii parameters; line numbers are added to the beginning of lines where none appear.
- default — The user must change to the correct subsystem before entering the RESEQ command.

  BASIC programs are resequenced under the BASIC subsystem: statements with line number references are updated. Note that language errors in the BASIC program may cause faulty resequencing.

  FORTRAN Extended programs are resequenced under the FTNTS subsystem; statement numbers and references to statement numbers are not changed.

All parameters supplied in the RESEQ command are order-dependent (must be entered in the order they appear in the command). Omitted parameters must be indicated by a comma if other parameters are to follow. For example, the command

```
RESEQ,50,,T
```

adds line numbers to the primary (text) file without checking for existing line numbers. Thus, the first line is numbered 00050 and each succeeding number is incremented by 10 (default value). If the T parameter was not specified in the command, the two preceding commas would not have been necessary.

If the starting line number (nn parameter) is greater than five digits in length or the increment value (ii parameter) causes the line numbers generated to exceed five digits in length, the message

```
LINE NUMBER LIMIT EXCEEDED.
```

is returned to the terminal. In addition, the message

```
RESEQ NUMERIC PARAM ERROR.
```

is returned if the user enters a nonnumeric value for either the nn or the ii parameters.

JOB SUSPENSION

A user may suspend his job at any time during program execution (RUN command issued). If an ASCII code terminal is being used, this is done by pressing the interrupt key (generally labeled INT, INTRPT, or BREAK). If the program is actively transmitting output to the terminal, the I key may also be used to suspend the job. If a correspondence code terminal is being used, only the ATTN key may be used to suspend the job. The system responds:

```
*INTERRUPTED*
```

A job can also be suspended automatically by the system when:

- The job has exceeded its time limit (the message TIME LIMIT is provided). The time limit is initially set to 100 octal seconds at log-in although the user may change this value through use of the SETTL command (refer to description in section 4).

- A successful recovery has been performed (refer to Recovery later in this section).

If the job was suspended by the user (interrupted) or by the system after a successful recovery, the user may perform one of the following.

- (CR) — Continue (transmit current output).† If the program was transmitting output to the terminal when it was suspended, the remaining output is transmitted when this response is made. Note that a portion of the output may be lost. When output is complete, program execution continues.

- P (CR) — Proceed (discard current output). If the program was transmitting output to the terminal when it was suspended, the remaining output is discarded and program execution continues when this response is made.

†If the TAPE command is in effect (tape mode), carriage return is ignored when entered on an empty input line. P (CR) must be issued to resume execution.

- STOP (CR) — Terminate the job.
- other — Terminates the job.

If the job was suspended because it exceeded its time limit (TIME LIMIT message issued), the user may respond as follows:

- T,nnnnn (CR) — Increment central processor time limit by nnnnn (number of octal seconds). When a job exceeds its time limit, this response is required to continue job execution. Any other input will terminate the job.

The following example illustrates the use of the I key to interrupt a program (note that the I key is effective only for ASCII code terminals).

```
LNH
00100 REM TEST FOR I KEY
00110 FOR I=1 to 100
00120 PRINT I
00130 NEXT I
00140 INPUT A,B
00150 PRINT A,B
00160 END
READY.

RNH
   1
   2
   3
   4
   5
   <-------------------------------  The job is interrupted by the I key.
*INTERRUPTED.*
   <-------------------------------  The user continues by pressing the carriage
   25                                return key (some output was lost).
   26
   27
   28
   29
   30
   31
   <-------------------------------  The job is again interrupted by the I key.
*INTERRUPTED.*
P  <-------------------------------  The user continues execution of the job by
                                     entering P (CR) (all output generated but
? 2,7                                not yet transmitted to the terminal when
   2                    7            the interrupt occurred is discarded).

SRU          0.105    UNTS.
```

In the following example, a program exceeds its allocated time limit. The program then runs to completion when a new time limit is entered.

```
      .
      .
RECOVER/SYSTEM:   FTNTS,OLD,NM
READY.

LNH

00100 PROGRAM T(OUTPUT)
00110 DO 6 I=1,2500
00120 DO 6 J=1,4000
00130 A=1
00140 6 CONTINUE
00150 END
READY.

SETTL,10

SRU        0.001 UNTS.
READY.

RNH

TIME LIMIT.
T,1000

 END.

SRU       15.421 UNTS.

RUN COMPLETE.
```

RECOVERY

During job processing, recovery may be necessary when:

- The terminal has been accidentally disconnected from the system.
- A system malfunction has occurred which requires a restart.

The user is placed in recovery state whenever he is disconnected from the system without logging off, providing that he is not already in recovery state. The user is given 10 minutes to initiate recovery.

In order to recover, a user completes the log-in sequence to the point where the system requests RECOVER/SYSTEM: (or RECOVER/CHARGE:). For example:

```
75/10/02.      16.07.24.
KRONOS TIME SHARING SYSTEM - VER. 2.1.1
USER NUMBER:  USER123,ABCDEFG <----------User number and password entered
TERMINAL:  15,    TTY <-------------------------Terminal number/type identifier
RECOVER/SYSTEM:
```

In response, the user enters:

```
RECOVER,nnn (CR)
```

Where:

nnn — Terminal number being used when the failure occurred. This number was indicated when the user initially logged-in (refer to preceding example). If the same terminal number is indicated when the user logs-in to recover, this parameter is not required.

The RECOVER command is valid only when entered in response to the RECOVER/SYSTEM: or RECOVER/CHARGE: request during the log-in sequence. If the user's terminal number is not the same as before the failure, the previous terminal number must be entered with the RECOVER command (nnn parameter). This may occur when the failure is due to a dropped line connection. In this case, a different terminal number may be assigned at log-in because of telephone switching rotaries. If the RECOVER command is not entered where indicated, the user may log-off, log-in again, and then enter the RECOVER command correctly.

If the system responds RECOVERY IMPOSSIBLE, one of the following conditions has occurred.

- No record was found of the specified user being logged-in on the given terminal number within the past 10 minutes.
- The user's system information was incorrect due to system malfunction.

If the user receives this message, he should check the terminal number he has entered. If he finds he has not given the correct number, he may reenter RECOVER with the correct terminal number.

If recovery is successful, the system responds:

```
RECOVERED - ENTER (CR) TO CONTINUE
LAST COMMAND - nnnn
NEXT OPERATION -
yyyyy
```

nnnn — Name of last command processed. If source code was being entered, the word SOURCE is output.

yyyyy — One of the following messages:

Message	Description
IDLE.	No activity.
INPUT REQUEST.	Job is awaiting input data.
OUTPUT AVAILABLE.	Output data is available.
EXECUTING.	Job is active with no output.
REENTER COMMAND.	Last command was not processed. The user should reenter the command.
REENTER INPUT.	The last line of input data was lost.
SORT.	The primary file is waiting to be sorted.

If the message FILES LOST is also issued, one or more working files was lost. This would occur only on system malfunction.

The user must then issue a carriage return to restart his job. If the job was previously transmitting output, the output data normally resumes a few lines prior to where the interruption in service occurred.

In general, recovery in the KRONOS system has been designed to provide minimum inconvenience to the user with a maximum of security. It should be realized, however, that under certain conditions recovery of user information cannot be perfect. The following are two examples of these circumstances.

- In some cases, a few lines of output may be lost when a phone line is disconnected just as a job is being restarted to generate more output. This actually occurs before current output is exhausted.

- If a phone line is disconnected when a user is entering source code, he can normally expect to lose the last few lines of text he has entered.

Whenever the user and terminal numbers of a user who must be placed in the recovery state match the user and terminal numbers of a user already in the recovery state, the user who was originally in the recovery state remains there and the other user is logged off.

An example of a normal recovery is:

```
75/10/02.     16.10.02
KRONOS TIME SHARING SYSTEM - VER. 2.1.2.
USER NUMBER:  USER123
PASSWORD
■■■■■■■■■
TERMINAL:  16,  TTY
RECOVER/SYSTEM:  RECOVER,12
RECOVERED - ENTER (CR) TO CONTINUE
LAST COMMAND - RUN
NEXT OPERATION -
IDLE.
```

The user types (CR) to continue job processing.

The following example illustrates a sample recovery.

```
72/10/02.     16.12.08.
KRONOS TIME SHARING SYSTEM - VER. 2.1.2.
USER NUMBER:  USER123,ABCDEFG
TERMINAL:  15,  TTY
RECOVER/SYSTEM:  FORTRAN,OLD,AA
READY.

LNH

10 PROGRAM AA(INPUT,OUTPUT)
20 PRINT 4
30 4 FORMAT(*ENTER 2 NUMBERS*)
40 READ,A,B
50 C=A*B
60 PRINT 5
70 5 FORMAT(*A X B=*)
80 PRINT,C
90 END
READY.
```

```
RNH

ENTER 2 NUMBERS
```
← The user is disconnected and logs in again.

```
75/10/02.     16.13.14
KRONOS TIME SHARING SYSTEM - VER. 2.1.
USER NUMBER:  USER123,ABCDEFG
TERMINAL:  15,  TTY
RECOVER/SYSTEM:  RECOVER
RECOVERED - ENTER (CR) TO CONTINUE.
LAST COMMAND-RNH
NEXT OPERATION -
INPUT REQUEST.
```
← The user enters (CR) to continue.

```
? 4,5
A X B=
                20.00
END.

SRU             0.004 UNTS.
RUN COMPLETE.
```

This example illustrates another recovery situation.

```
STATUS
TERMINAL:  32,  TTY
SYSTEM - BASIC
FILE NAME:  BB
STATUS - IDLE

LNH

10 REM THIS IS A TEST
20 REM PRINT B
30 PRINT "PRINT B"
40 LET B=1+3
50 PRINT B
60 END
READY.
```
← The user is disconnected.

```
75/10/04.     15.16.23.
KRONOS TIME SHARING SYSTEM - VER. 2.1.
USER NUMBER:  USER123,ABCDEFG
TERMINAL:  32,  TTY
RECOVER/SYSTEM:
```
← The user is disconnected before recovering. This is processed as a normal log-off.

```
75/10/04.     15.17.20.
KRONOS TIME SHARING SYSTEM - VER. 2.1.2.
USER NUMBER:  USER123,ABCDEFG
TERMINAL:  32,  TTY
RECOVER/SYSTEM:   RECOVER,32 <----The user enters the RECOVER command.
RECOVERED - ENTER (CR) TO CONTINUE.
LAST COMMAND-LNH
NEXT OPERATION -
IDLE.
   <-------------------------------- The user enters (CR) to continue.
LNH

10 REM THIS IS A TEST
20 REM PRINT B
30 PRINT "PRINT B"
40 LET B=1+3
50 PRINT B
60 END
READY.
```

REMOTE JOB ENTRY 8

Remote job entry is a feature of KRONOS that enables a validated, time-sharing user to submit batch job deck images to the system for processing. The batch job image can be created interactively at the time-sharing terminal and then submitted to the local batch queue by entering the SUBMIT command. The user is no longer interactive with the job once it has been submitted. Instead, the job is treated in the same manner as a job submitted from the card reader at the local computer site or from a remote batch terminal (EXPORT/IMPORT). The batch job image can consist of control card images, KRONOS control language statements, program statements, and data. The structure of the batch job image must also be the same as its counterpart on cards (job deck). The user can specify how the job is to be structured through use of the reformatting directives described in this section. Refer to the KRONOS Reference Manual for information concerning job structure and a description of the batch control cards and control language statements available.

The format of the SUBMIT command is:

SUBMIT(lfn, q, NR)c

Where:

lfn — Name of file to be submitted to the system for processing as a deferred batch job. If lfn is omitted, the primary file name is assumed. In this case, the primary file is sorted before being submitted. Permanent files may also be submitted but first must be associated with the job via the OLD, GET, or ATTACH command.

q — Specifies disposition of job output as follows:

- B — Job output is disposed to local batch queue to be printed, punched, etc., at the central site.
- N — Job output is disposed to local batch queue, but output is dropped at job termination (default value).
- E — Job output is disposed to EXPORT/IMPORT queue for printing at a remote batch terminal.

NR — No rewind option; inhibits rewind of file specified by reformatting directive cREAD (refer to Reformatting the Submit File later in this section). If omitted, file specified by cREAD directive is automatically rewound.

c — Escape character used to identify reformatting directives in the file to be submitted (lfn). If omitted, the system assumes c=/.

After the job has entered the local batch queue, the system responds:

```
hh.mm.ss.jobname
READY.
```

Where:

hh.mm.ss.	Time that the job entered the batch queue (hours.minutes.seconds.)
jobname	Job identification

The job identification is entered with the ENQUIRE,an=jobname command to obtain the status of a job once it has entered the system. Refer to the description of the ENQUIRE command in section 4 for additional information.

REFORMATTING THE SUBMIT FILE

The submit file (lfn in the SUBMIT command) is a file which contains the batch job image submitted to KRONOS for processing. The reformatting directives described in this section are provided to aid the user in preparing the submit file. When the SUBMIT command is entered, the submit file is reformatted according to the directives that appear in the file and is then placed in the batch input queue awaiting execution. Thus, several of the directives are provided to format the submit file to meet the structural requirements of a batch job (refer to section 3 in the KRONOS Reference Manual).

Each line in the submit file preceded by an escape character is recognized by KRONOS as a reformatting directive. The escape character to be used must be defined in the SUBMIT command (/ by default). Throughout this description, the letter c, preceding a directive, denotes the escape character. Reformatting directives may be interspersed throughout the submit file as long as transparent mode is not in effect. Transparent mode is selected by the cTRANS directive and requires that the user observe special rules when inserting subsequent directives into the file (refer to the description of cTRANS and cNOTRANS directives).

It is important to note that reformatting directives will not be processed unless the first line of the submit file contains the cJOB directive. In addition, the first two card images following the cJOB directive (second and third lines of the submit file) must be a JOB and USER card, respectively. All following information is determined by the user. Thus, the first three lines of a submit file that is to be reformatted before processing should be:

```
00100 cJOB
00110 JOBNAME,...  <-------------------- JOB card image
00120 USER,...     <-------------------- USER card image
```

NOTE

The JOB and USER cards are required in all batch jobs and must be present even though the cJOB card is omitted.

Although text mode may be used to create a submit file, it is not necessary since each line may or may not have a leading line number. The cSEQ and cNOSEQ directives are used to determine, during reformatting, if line numbers in the submit file will be retained. Therefore, it is a simple matter to include line numbers on the entire submit file and specify which line numbers are to be removed during reformatting. This is especially useful if the submit file contains a BASIC program where line numbers are a requirement of the language.

The reformatting directives available are described as follows:

cJOB — Indicates that the submit file is to be reformatted and selects the following default reformatting directives. The default directives remain in effect until specified otherwise.

cNOTRANS (disabled by cTRANS)
cSEQ (disabled by cNOSEQ)
cPACK (disabled by cNOPACK)

The cJOB directive must be the first line of the submit file. If omitted, the file is not reformatted.

cEOR — Indicates that an end-of-record mark is to be placed at this point in the submit file during reformatting

cEOF — Indicates that an end-of-file mark is to be placed at this point in the submit file during reformatting

cSEQ — Indicates that line numbers will be removed from all subsequent lines during reformatting (default value)

cNOSEQ — Reverses the effect of the cSEQ directive. No attempt is made to remove leading line numbers from subsequent lines.

cPACK — Indicates that all succeeding internal EOR and EOF marks be removed during reformatting (default value). This directive applies only to internal EOR and EOF marks that currently exist. The cEOR and cEOF reformatting directives are not affected.

cNOPACK — Reverses the effect of the cPACK directive. Requests the system not to discard succeeding internal EOR and EOF marks that currently exist.

cTRANS — Indicates transparent mode. When encountering this directive during reformatting, KRONOS checks the next line of the submit file for an additional reformatting directive. If one exists, it is processed and the following line is checked. This continues until a line that does not contain a reformatting directive is encountered. Transparent mode is then selected and all directives that exist on subsequent lines are ignored until an internal EOR or EOF is encountered (not cEOR or cEOF). The cPACK and cNOPACK directives determine if the internal EOR or EOF

will be retained. The line following the internal EOR or EOF mark is then checked for a reformatting directive. If one exists, it is processed and the following line is checked. All directives are processed until a line that does not contain a reformatting directive is encountered. This causes transparent mode to be reset unless a cNOTRANS directive was encountered. This process continues until either the end of the submit file is reached or until a cNOTRANS directive following an internal EOR or EOF is encountered.

The cTRANS directive is typically used in conjunction with the cREAD directive. This allows the user to copy the contents of an existing file into the submit file at the location of the cREAD directive. Because the file is read in transparent mode, no check for reformatting directives is attempted until an internal EOR or EOF is encountered. Note that the cREAD directive must follow the cTRANS directive and must be located before the first succeeding line that is not a reformatting directive. If not, transparent mode is selected before the cREAD directive is encountered, and the cREAD will be ignored.

The cSEQ or cNOSEQ directive in effect before transparent mode was selected has no effect upon the submit file or the file being read (cREAD) while transparent mode is in effect. Note, however, that the cPACK or cNOPACK directive in effect before transparent mode was selected remains in effect after it is selected.

cNOTRANS — Reverses the effect of the cTRANS directive and informs the system that the submit file is to be examined on a line-by-line basis. All directives encountered in the submit file while the cNOTRANS directive is in effect are processed. This directive is initially selected by default and remains in effect until a cTRANS directive is encountered in the submit file.

Caution should be observed in the placement of this directive in the submit file. If transparent mode is selected, this directive can possibly be ignored unless it immediately follows either a cREAD directive in the submit file or an internal EOR or EOF mark.

cREAD, lfn — Requests that the system read the entire contents of the specified file (lfn) and insert that file in place of the cREAD directive in the submit file, during reformatting. If the file to be read is not currently attached to the user, KRONOS automatically attempts a GET and then an ATTACH on the file. If lfn is not specified in the directive, TAPE1 is assumed. If the file specified cannot be found, the message

NO READ FILE - lfn

is issued to the job dayfile, and the job is terminated. If the read file is found to be busy (direct access files only), the message

READ FILE BUSY - lfn

is issued to the job dayfile and the job is terminated. The file specified by lfn in the cREAD directive is automatically rewound before the read operation unless the NR parameter is specified in the SUBMIT command. In this case, the cREWIND directive must precede the cREAD directive in the submit file if it is desired to rewind file lfn before the read operation begins. KRONOS returns all files specified in cREAD directives before completion of the job.

If the cPACK directive is in effect at the time of the read, all internal EOR and EOF marks will be removed. If the cNOPACK directive is in effect, all internal EOR and EOF marks are read into the submit file in the proper position during reformatting.

Unless transparent mode is in effect when file lfn is read, each line of that file is also checked for a reformatting directive. Any directives contained in the file, except another cREAD, are processed. The cREAD directive cannot be nested. In addition, any directives in effect before the cREAD directive is processed remain in effect for the file being read, unless transparent mode is selected. Then, only the cPACK or cNOPACK directives remain in effect for the file being read. Moreover, only those directives that immediately follow an internal EOR or EOF in the file being read are processed. If the file to be read is a binary file, it is recommended that the cTRANS directive be used. This is to ensure that binary data will not be mistaken for a reformatting directive. If used, the cTRANS directive should immediately precede the cREAD directive in the submit file.

cREWIND, lfn — Requests that the system rewind file lfn to the beginning-of-information (BOI). If lfn is not supplied, TAPE1 is assumed. This directive is required only if the NR parameter is included in the SUBMIT command. Otherwise, file lfn is automatically rewound.

This directive is used in conjunction with the cREAD directive. Thus, if it is desired to rewind a file before the read operation begins, this directive must precede the cREAD directive in the submit file.

c_1EC=c_2 — Indicates that the escape code character is to be changed from c_1 (current escape code) to c_2 (new escape code). The new escape code is used to recognize all subsequent reformatting directives until further change.

ERROR PROCESSING

If the user determines that an error occurred during processing of his job, he may reference a listing of the job dayfile as an aid in identifying the cause of the error. The job dayfile contains a record of the job processing activity and is disposed to the local batch queue or the EXPORT/IMPORT queue for printing when the job is terminated. However, all output is normally dropped at job termination when a batch job image is submitted from a time-sharing terminal. This includes the dayfile output as well as the job's output. In this event, the user can make provisions within his job to save the contents of the dayfile if an error in processing occurs. This is done by including the following control card images at the end of the job's control record (first record of the submit file).

```
.
.

00562 EXIT.
00563 DAYFILE,lfn.
00564 REPLACE(lfn)

.
.
```

When an error condition occurs during job processing, the system searches the job's control record for an EXIT card. If an EXIT card is found, the error condition is cleared and the control card images that follow are processed. In this case, the contents of the job dayfile is copied to the working file specified by lfn. That file is then saved for future reference.

If an error does not occur during job processing, the EXIT card indicates where to terminate normal control card processing.

BATCH SUBSYSTEM 9

The BATCH subsystem extends the capability of the terminal user by allowing him to enter batch control card images as well as time-sharing commands. The control card images are entered and processed one at a time, and all output is returned to the terminal unless otherwise specified. Refer to the KRONOS Reference Manual for a description of the batch control cards available and information concerning their use.

To enter the BATCH subsystem, the user types:

BATCH, nnnnn

Where:

nnnnn — Amount of field length required for the control card images to be entered. If nnnnn is not specified, a default value of 20000B is assumed. Note that an RFL control card image can be entered at any time to specify a new field length.

Many of the system utilities and file manipulation control cards (for example, COPYBF and SKIPF) run at a predefined field length rather than the field length specified in the BATCH command. However, all other control cards use the field length specified in the BATCH command or by a subsequent RFL control card. The field length required for the product set control cards (for example, COMPASS) is listed in the KRONOS Reference Manual.

The system responds:

```
$RFL, nnnnnn.      (nnnnn is the specified field length)
/
```

The user may then enter any valid batch control card image or time-sharing command. Time-sharing commands cannot be abbreviated under the BATCH subsystem; the entire command or control card must be entered to be valid. The user may also end each control card with a period, but this is not required since the system automatically terminates each line of batch input with a period. If the command or control card is not valid or is entered incorrectly, the system responds:

ILLEGAL CONTROL CARD.

All the commands described as permanent file commands are processed as batch commands. When a command is treated as a batch command, its operation depends upon a previously entered USER control statement.

All commands that are processed in batch mode can be included in a procedure file. Those commands that are processed by the time-sharing executive cannot be included in a procedure file. However, if the user desires that a time-sharing command be processed as a control card, he may enter the following command (available only under the BATCH subsystem):

X, ccccccc

This command allows the user to force a valid time-sharing command (ccccccc), normally interpreted by the time-sharing executive, to be processed by the system as a control card. For example, to call the BASIC compiler, the user enters:

X, BASIC

rather than the command

BASIC

which would change the subsystem from BATCH to BASIC.

To exit from the BATCH subsystem, the user must type the entire name of one of the other subsystems (FTNTS, BASIC, EXECUTE, or NULL). It is important to note that the user may also change the subsystem unintentionally by entering the OLD command. The OLD command is used to obtain a different primary file, but may also select the subsystem that was active when the specified file was originally saved. Refer to the description of the SAVE command in section 5 for additional information.

NOTE

Although a user can enter any batch control card from a time-sharing terminal, the output produced by several of these cards is formatted for transmission to a line printer (136 characters per line). Through use of the L072 control card, the user can format the output for transmission to a time-sharing terminal (72 to 150 characters per line). Refer to the description of the L072 control card in section 5 of the KRONOS Reference Manual.

Although the user is not expressly prohibited from entering the user control statement to specify a different user number and password, the results may be unpredictable.

All permanent file time-sharing commands and batch commands are processed under the user number specified in the most recent user card entered. The time-sharing commands are processed under the log-in user number. If the limits and validations are different between the two user numbers, conflicts and inconsistencies may arise in the processing of the job.

GLOSSARY

Alphanumeric	A combination of intermixed alphabetic and numeric characters.
Answerback code	A character code which uniquely identifies a teletypewriter unit. If this code is present in the user's validation file, the user's activities are limited to this particular unit. The code is set on a drum within the TTY. KRONOS checks only the first ten characters.
APL	A Programming Language. An interactive interpreter available to the time-sharing user.
ASCII	American National Standard Code for Information Interchange; the 128-character set utilized by KRONOS for the terminal user.
Auxiliary device	Mass storage permanent file device used to supplement storage provided by the normal permanent file devices associated with the system (refer to family device).
BASIC	Beginner's All-purpose Symbolic Instruction Code; an elementary programming language available to the KRONOS user.
Catalog	The list of permanent files saved under a particular user number; this list also contains information about the permanent files.
Compile	To translate a program from a higher level programming language (for example, FORTRAN or BASIC) into machine instructions.
Control card	Instruction to the operating system to perform a specified task or utility.
Direct access file	A permanent file which can be attached to the user's job. All changes to this file are made on the file itself rather than to a working copy of the file. (Refer to indirect access file.)
Display code	An internal code set used by CDC CYBER 70 series and 6000-series computers to represent alphanumeric and special characters. (Refer to Table B-2 in appendix B.)

EOF EOI EOR	End-of-file End-of-information End-of-record These represent the method by which files are defined and structured. Refer to the discussion in section 3.
Escape key	The ESC key on ASCII code terminals which allows the user to delete the current line of input.
Family device	Mass storage permanent file device associated with a specific KRONOS system. A family may consist of 1 to 63 logical devices. Normally, a system runs with one family of permanent file devices available. However, additional families may be introduced during normal operation. This enables the users associated with the additional families to access their permanent files via an alternate system.
Field length	Number of central memory words required to process a job.
File	Either a program or data, made up of one or more groups of data separated by EORs and EOFs and terminated by an EOI.
Flag	A character or bit that signals the occurrence or presence of a particular condition.
FORTRAN	The FORmula TRANslation compiler that converts algebraic and scientific problems into machine language instructions.
Indirect access file	A permanent file which is accessed by use of a working copy of the file rather than by working on the file directly.
Interactive	Job processing in which the user and the computer communicate with each other, rather than the user merely submitting his job, receiving output, and having no control over the job while processing occurs.
Library file	A permanent file residing under the special user number LIBRARY.
Local file	Any file assigned to a user while he is logged in at a terminal. Same as a working file.
Locked file	A file on which a user cannot write.
Logical record	A record whose length is a function of the information it contains rather than the device on which it is stored.

Master device	Mass storage device that contains the user's permanent file catalog, all his indirect access files, and all, part, or none of his direct access files.
Object code	Executable machine language instructions. An object code program need not be recompiled each time it is executed.
On-line	Equipment under direct control of the computer.
Order-dependent	Items which must appear in a specific order.
Order-independent	Items which need not be given in any specific order. Parameters may be order-independent. For example, the UN=usernum parameter in the GET command is order-independent.
Parameter	A variable that is given a specific value for a particular purpose or process.
Permanent file	A file which is created by a user, stored under his user number in the permanent file system, and is removed from permanent file storage only when specified by the user.
Permanent file family	Permanent files which reside on the family devices of a specific KRONOS system.
Physical record unit (PRU)	The standard unit of data storage defined by the device on which a file resides.
Port	The point at which a communication line is attached to the computer system.
Post radix	A letter following a numeral that indicates the base numbering system.
Prefix character	A character that has a special significance to a program or the operating system.
Primary file	Any working file defined with the OLD and LIBRARY command or initiated with the NEW command. The primary file is assumed to be the file on which most operations are performed unless another file is specified. The primary file is automatically sorted by the system and rewound after each use.
Private auxiliary device	Auxiliary device associated with a specific user. Only that user may create files on the device, although he may permit other users to access files which reside on the device.

Public auxiliary device	Auxiliary device available for access by all validated users knowing the correct packname. Note that additional validation is required to create or replace files on an auxiliary device (refer to LIMITS command in section 4).
Rollout file	A file containing a job (and system information) that has been temporarily removed from the main processing area of the system.
Rubout characters	Characters created by pressing the RUBOUT key on the terminal. These are considered null input by the system and are required in paper tape input.
Source code	Code input to the computer for later translation into executable machine language instructions (object code).
Swapping	The process of removing one job from central memory and replacing it with another job.
System file	A file that can be accessed only by a system program.
System Resource Unit	A unit of measurement of system usage. If a charge/project number is not used, the number of SRUs is a measurement of the CPU time, memory usage, and I/O activity for this terminal session since log-in. If a charge/project number is used, the number of SRUs is given for all sessions under that charge/project number.
System utilities	System programs used to perform system functions.
Time slice	The amount of CPU time a job is allowed to use before the system lowers its priority to allow other jobs to execute.
TTY	Teletypewriter.
Validation file	File containing validation information for all users (user numbers, passwords, resources allowed, etc.)
Working file	Any file assigned to a user while he is logged in at a terminal.

TERMINAL ERROR MESSAGES A

The following error messages may appear at the terminal.

Message	Description
APPEND ERROR.	An error was encountered when appending a file.
ARGUMENT ERROR.	An invalid CONVERT control statement option was specified.
BUFFER ARG. ERROR.	Address of information for permanent file manager was in error.
CATALOG OVERFLOW FILES.	User has exceeded the number of permanent files he is allowed.
CATALOG OVERFLOW SIZE.	User has exceeded the mass storage area allowed for indirect access permanent files.
CHARGE FILE BUSY.	File is not available for charge purposes; wait and try again.
CHARGE ILLEGAL AT THIS HOUR.	Project number used is not valid at this time of day.
CHARGE REQUIRED.	Charge number and project number are required to complete log-in sequence.
COMMAND NOT UNIQUE.	The characters supplied are not unique to one command.
CONTROL STATEMENT LIMIT.	The number of control statements processed has exceeded the limit for which the user was validated.
CONVERSION NOT FOUND.	The specified TS conversion table was not found.
CONVERSION NOT SPECIFIED.	Neither the TS nor 64 parameter was specified on the CONVERT control statement.
DEVICE UNAVAILABLE.	User is attempting to access files on a device not available for access.
DIRECT ACCESS DEVICE ERROR.	User is attempting to DEFINE or ATTACH direct access permanent files on a device not available for access.
EMPTY CATALOG.	No files exist for the options specified in a CATLIST command.

Message	Description
ERROR IN xxxxxx ARGUMENTS.	The command string given contained a syntax error. xxxxxx is the name of the system program issuing the diagnostic. This message is similar to ILLEGAL PARAMETER.
EOF ENCOUNTERED BEFORE TERMINATION.	An end-of-file was encountered on the convert input file before the specified record count was exhausted.
EOI ENCOUNTERED BEFORE TERMINATION.	An end-of-information was encountered before the specified record count was exhausted.
ERROR IN FILE TYPE.	File type specified was not recognized (for example, CT=SEMI instead of CT=SPRIV).
ERROR IN MODE.	Mode specified was not recognized.
ERROR IN PFILE ARGUMENTS.	An argument for permanent file request was illegal.
EXECUTE ONLY FILE.	The user has only an E permission level on the file.
FILE NAME CONFLICT.	The correct I/O file names specified are the same.
FILE NAME ERROR.	The file name must contain only alphanumeric characters.
FILE NAME MISSING.	A file name must be specified before a listing containing the permit information can be obtained with a CATLIST command.
FILE NOT SORTED.	Primary file has one or more lines with nonnumeric line numbers. (Use SORT, lfn command to obtain more detailed diagnostics.)
FILE TOO LONG.	Limit for length of indirect access permanent file has been exceeded or limit for length of a direct access file has been exceeded prior to attaching in write modify of append modes.
FILE TOO LONG TO SORT.	Primary file is too long to be sorted automatically by the system. (Enter SORT, lfn command to obtain more detailed diagnostics.)
FL TOO SHORT FOR PROGRAM.	Self-explanatory; increase field length with RUN command (or RFL card in BATCH subsystem).
ILLEGAL CHARGE.	Either charge or project number does not exist or the project number entered is not available to this user number.
ILLEGAL COMMAND.	The command name entered is not a valid command.
ILLEGAL LIST OPTION.	The list option specified in a CATLIST command is illegal.

Message	Description
ILLEGAL LOGIN.	The validation (VALIDUS) file for the permanent file family specified is not present in this system.
ILLEGAL PARAMETER.	May indicate: • Parameter is outside legal bounds. • Parameter is not one of the acceptable sets for the command. • Parameter referenced does not exist.
ILLEGAL USER ACCESS.	User has either attempted an operation that he has not been validated to perform or has exceeded the validation limits (refer to the LIMITS command in section 4).
ILLEGAL WRITE ON APPEND ONLY FILE.	User has attempted to write on a permanent file with append-only permission to the file.
ILLEGAL WRITE ON MODIFY ONLY FILE.	User has attempted to write on a direct access file with modify-only permission to the file.
ILLEGAL WRITE ON READ ONLY FILE.	User has attempted to write on a permanent file with read-only permission to the file.
LOCAL FILE LIMIT.	User's local file limit was exceeded during an attempt to define a new file or retrieve an existing permanent file.
lfn EMPTY.	The file contains no data.
lfn NOT ON MASS STORAGE.	The user has attempted to save a file that does not reside on mass storage.
MESSAGE† LIMIT.	The number of job dayfile messages issued by the job has executed the limit for which the user was validated. Message functions issued by compilers or applications programs that run at the user job's control point are also counted as user dayfile messages.
nnnnnn RECORDS CONVERTED.	nnnnnn records were converted.
NO CONNECT TIME AVAILABLE.	Terminal connect time allowed under this project number has been expended.
NO CPU TIME AVAILABLE.	Central processor time allowed under this project number has been expended.
NO PRIMARY FILE.	The user has attempted an operation which requires a primary file when there was none defined.

Message	Description
OUTPUT FILE LIMIT.	The number of files disposed to output queues by the user's job has exceeded the limit for which the user was validated.
pfn ALREADY PERMANENT.	The specified file already exists (use REPLACE).
pfn BUSY.	The file to be attached is currently assigned to another user in conflicting mode.
pfn NOT FOUND.	The permanent file specified does not exist. If this message occurs in response to the CATLIST command, pfn may specify a user number.
PERMISSION NOT GRANTED.	The user does not have WRITE or APPEND permission on the specified permanent file.
PF UTILITY ACTIVE.	Permanent file utility operation is currently taking place. Permanent file commands (OLD, LIB, GET, etc.) are aborted. Control card images entered in the BATCH subsystem are placed on recall until the utility operation is complete.
PFM ILLEGAL REQUEST.	An illegal request was issued to the permanent file manager.
PRU LIMIT	The job's mass storage PRU limit was exceeded during preparation of a local copy of an indirect access file or by write extension of an existing file by the job.
RECORD SIZE EXCEEDS 500.	The maximum line allowed for converting a record is 500.
RESERVED FILE NAME.	File name is reserved by the system (for the user's protection).
SYSTEM CLOSED, SORRY	The system is not available for use by terminals.
TOO MANY PARAMETERS.	The user has submitted more parameters than the command allows.

ERROR MESSAGES FOR SYSTEM MALFUNCTIONS

The following messages indicate system error conditions:

Message	Description
DATA BASE ERROR.	An error in the structure of the validation file has been detected. Report this error to the computer center immediately.

Message	Description
ROLLIN FILE BAD.	If this message is received, the user should log-off immediately. If he desires to continue, he may log-in again. The user should report this malfunction to the computer center and supply all terminal output since log-in, if possible.
SYSTEM ERROR.	This error message should never appear. If it does, the user should report the malfunction to the computer center and supply all terminal output since log-in, if possible.

DIAL-IN PROCEDURE AND CHARACTER SET B

This appendix presents an explanation of how to connect an interactive terminal to the KRONOS Time-Sharing System. Also included are illustrations of typical keyboard configurations for terminals using either standard or APL print, the function of certain control keys when connected to KRONOS. Table B-2 lists the 64-character set.

The procedure to connect the terminal to the system depends upon whether a data set or an acoustic coupler is being used to establish the connection. Both procedures are included in this appendix. Note that some terminals (most teletypewriter models) can include a data set or an acoustic coupler as an integral part of the terminal.

It is recommended that the user read the instruction manual, generally supplied with the terminal, to become familiar with the operation of the terminal before connecting to KRONOS (that is, how to apply power, load paper, etc.).

CONNECTING TERMINAL USING A DATA SET

If the data set is included as part of the terminal, proceed as follows:

1. Apply power to the terminal (refer to appropriate instruction manual). A dial tone should be audible when power is applied.
2. Dial the correct phone number. A short high pitched signal indicates that the call has been answered.

If the data set is a stand-alone unit, proceed as follows:

1. Apply power to the terminal (refer to appropriate instruction manual).
2. Remove the data phone receiver from its cradle.
3. Press the TALK key on the data set to obtain a dial tone.
4. Dial the correct phone number. A constant high pitched signal indicates that the call has been answered.
5. Press the DATA key on the data set (indicator in key should light), and replace the receiver in its cradle.

CONNECTING TERMINAL USING AN ACOUSTIC COUPLER

1. Apply power to the terminal (refer to appropriate instruction manual).
2. Turn on the acoustic coupler (not required if the coupler is included as part of the terminal).
3. Using a standard telephone, obtain a dial tone and dial the correct phone number. A constant high pitched signal indicates that the call has been answered.
4. Place the receiver in the acoustic coupler.

Figure B-1 illustrates a typical terminal keyboard with standard print.

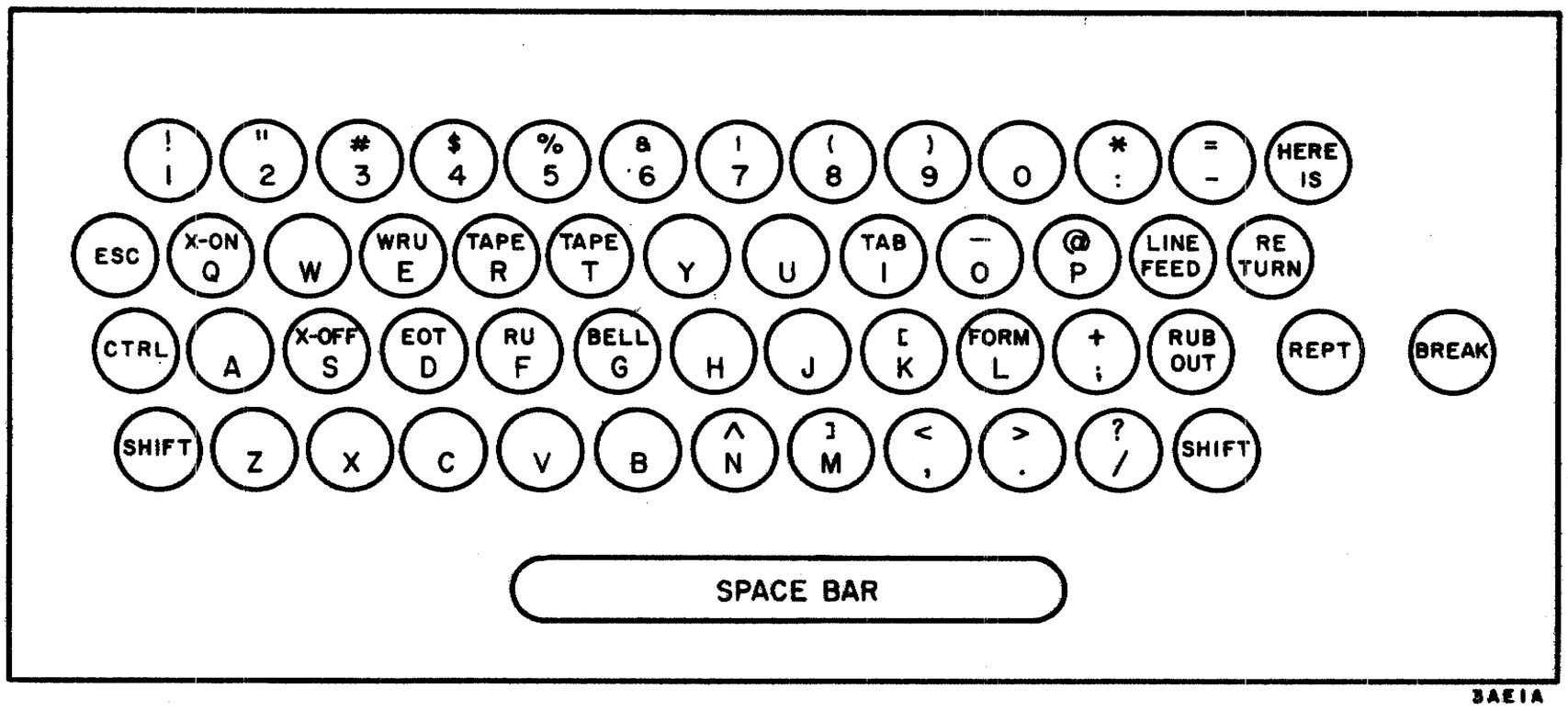

Figure B-1. Teletype Model 33 Keyboard

Figure B-2 illustrates a typical terminal keyboard with APL print.

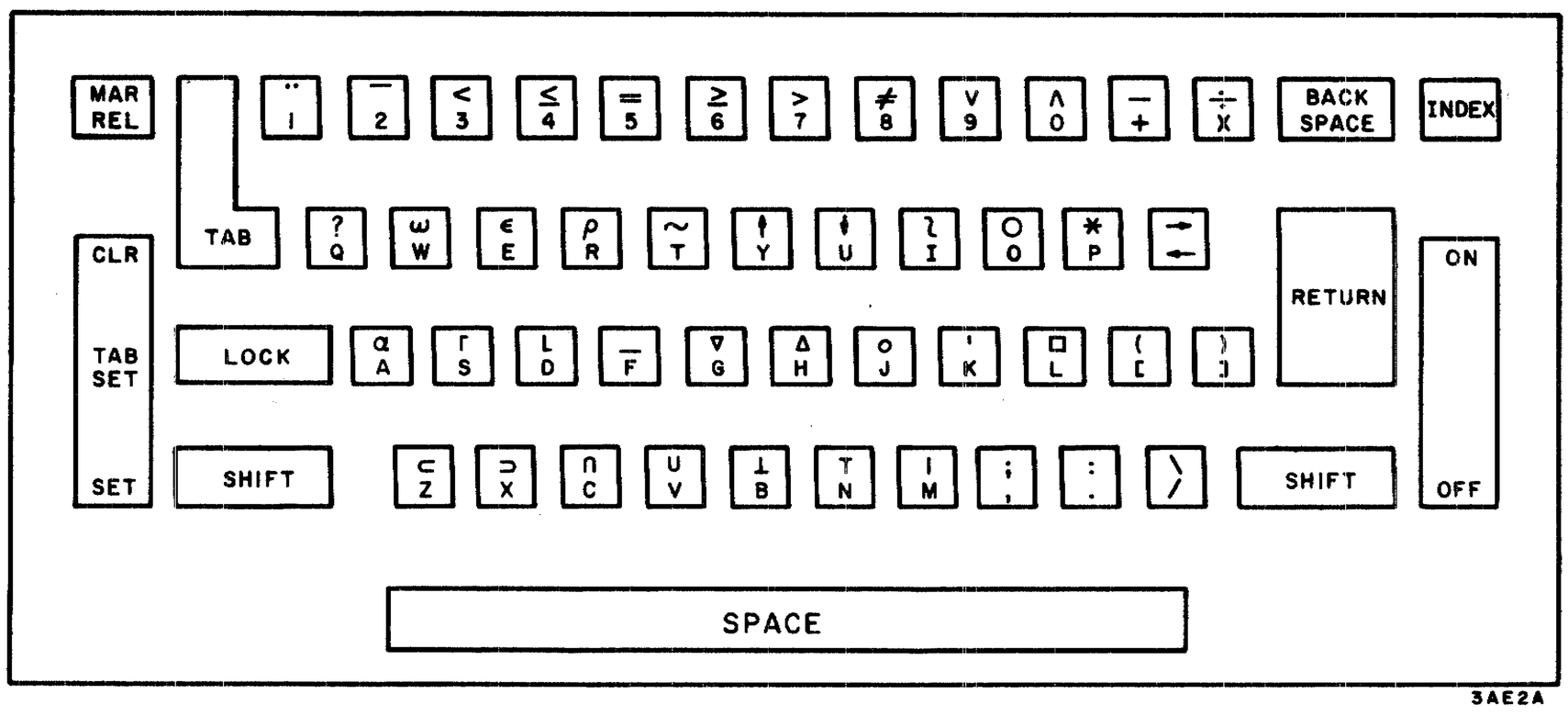

Figure B-2. Novar 5-41 Keyboard (APL Language)

Table B-1 illustrates the function of certain terminal control keys when connected to the KRONOS Time-Sharing System.

TABLE B-1. TIME-SHARING TERMINAL CONTROL KEYS/FUNCTION

Function (function of control key)	Control Keys						
	ASCII Code Terminals (typical)				Correspondence Code Terminals (typical)		
	Teletype 33/35	Teletype 37/38	CDC 713 Display	Memorex 1240	IBM 2741	Novar 5-41	Datel 30
Delete current input line	ESC	ESCAPE	ESCAPE	ESC	ATTN	ATTN	ATTN
Backspace	Underline or Backarrow	Underline	Underline or Backspace	Underline	BACK SPACE	BACK SPACE	BACK SPACE
Enter text mode	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command
Exit text mode	BREAK† or ETX (CTL X)	INTRPT† or ETX (CTL X)	BREAK† or ETX (CTL X)	INT†	ATTN†	ATTN†	ATTN†
Interrupt program (during output)	BREAK or I key	INTRPT or I key	BREAK or I key	INT or I key	ATTN	ATTN	ATTN
Interrupt program (no output)	BREAK	INTRPT	BREAK	INT	ATTN	ATTN	ATTN
Terminate program (during output)	S key	S key	S key	S key	ATTN	ATTN	ATTN
Terminate program (no output)	STOP command	STOP command	STOP command	STOP command	STOP command	STOP command	STOP command
Punch leader on paper tape	RUB OUT	DELETE	NA	NA	NA	NA	NA

† Effective only when entered in first character position of line (empty input line)

CENTRAL MEMORY CHARACTER EQUIVALENTS FOR DATA INPUT

The manner in which characters entered from the terminal are interpreted by the system depends upon whether the user has specified that the characters belong to the full ASCII set (refer to the description of the ASCII command in section 4). For example, if the user enters the following characters to be mapped into the full ASCII set (128 characters recognized):

aAbBcCdDeE

the central memory equivalent (in display code) is as follows:

59		47		35		23		11		0 ← Bit position (60-bit central memory word)
76	01	01	76	02	02	76	03	03	76	
04	04	76	05	05	55	00	00	00	00	

However, if the characters are mapped into the subset of the ASCII character set used during NORMAL operation (61 characters recognized), the central memory equivalent (in display code) is as follows:

59		47		35		23		11		0
01	01	02	02	03	03	04	04	05	05	

Characters entered under control of the BINARY command are not translated into display code. It makes no difference if the ASCII command is in effect (128 characters recognized) or the characters are entered during NORMAL operation (61 characters recognized). However, each 8-bit character is weighted by a value of 4000B. For example, if the user enters the following characters and the BINARY command is in effect:

aAbBcCdDeE

the central memory equivalent is as follows:

59		47		35		23		11		0
4341		4101		4342		4102		4143		
4303		4344		4104		4145		4305		

The characters illustrated are entered from an ASCII code terminal although a correspondence code terminal may also be used.

CENTRAL MEMORY CHARACTER EQUIVALENTS FOR DATA OUTPUT

The previous discussion applies but with the following notes.

Data output is in either a 64/63- or 128-character set, depending upon whether the terminal is in NORMAL or ASCII mode, respectively. When the terminal is in NORMAL mode, codes 74 and 76 represent data rather than escape codes. In ASCII mode, a 74 or 76 code is treated as the beginning of an escape sequence. Under 64-character set ASCII, the 7404 code maps to : (colon). Under 63-character set ASCII, the 63 code maps to : (colon), while the 7404 code maps to % (percent or the ÷ on CORAPL and MEMAPL terminals). In either 63- or 64-character set ASCII, the code 00 does not exist.

TABLE B-2. 64-CHARACTER SET

ASCII CODE TERMINAL†				CORRESPONDENCE CODE TERMINAL††				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	
:	072	:	276	:	153	:	121	00
A	101	A	341	A	171	A	171	01
B	102	B	342	B	166	B	166	02
C	303	C	143	C	172	C	172	03
D	104	D	344	D	052	D	052	04
E	305	E	145	E	112	E	112	05
F	306	F	146	F	163	F	163	06
G	107	G	347	G	043	G	043	07
H	110	H	350	H	046	H	046	10
I	311	I	151	I	031	I	031	11
J	312	J	152	J	103	J	103	12
K	113	K	353	K	032	K	032	13
L	314	L	154	L	106	L	106	14
M	115	M	355	M	141	M	141	15
N	116	N	356	N	122	N	122	16
O	317	O	157	O	105	O	105	17
P	120	P	360	P	013	P	013	20
Q	321	Q	161	Q	133	Q	133	21
R	322	R	162	R	051	R	051	22
S	123	S	363	S	045	S	045	23
T	324	T	164	T	002	T	002	24
U	125	U	365	U	062	U	062	25
V	126	V	366	V	061	V	061	26
W	327	W	167	W	165	W	165	27
X	330	X	170	X	142	X	142	30
Y	131	Y	371	Y	147	Y	147	31
Z	132	Z	372	Z	124	Z	124	32
0	060	0	060	0	144	0	144	33
1	261	1	261	1	040	1	040	34
2	262	2	262	2	020	2	020	35
3	063	3	063	3	160	3	160	36
4	264	4	264	4	004	4	004	37
5	065	5	065	5	010	5	010	40
6	066	6	066	6	130	6	130	41
7	267	7	267	7	150	7	150	42
8	270	8	270	8	070	8	070	43
9	071	9	071	9	064	9	064	44
+	053	+	055	+	023	+	067	45
-	055	-	275	-	067	-	067	46
*	252	*	120	*	070	*	013	47
/	257	/	257	/	007	/	007	50
(	050	(	053	(	064	(	153	51
)	251	)	252	)	144	)	111	52
$	044	$	176	$	004	α	171	53
=	275	=	245	=	023	=	010	54

3AE4A

† THE OCTAL CODES LISTED FOR ASCII CODE TERMINALS ARE SHOWN WITH EVEN PARITY (NORMAL)

†† THE OCTAL CODES LISTED FOR CORRESPONDENCE CODE TERMINALS ARE SHOWN WITH ODD PARITY (NORMAL)

TABLE B-2. 64-CHARACTER SET (Cont'd)

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	
(SPACE)	240	(SPACE)	240	(SPACE)	100	(SPACE)	100	55
,	254	,	254	,	073	,	073	56
.	056	.	056	.	121	.	121	57
#	243	¨	041	#	160	¨	040	60
[	333	[	273	¼	001	[	153	61
]	335	]	072	½	001	]	111	62
%	245	÷	173	%	010	÷	023	63
"	042	≠	050	¨	111	≠	070	64
_	137†	_	306	_	067	_	163	65
!	041	∨	251	¢	130	∨	064	66
&	246	∧	137	&	150	∧	144	67
'	047	'	113	'	111	'	032	70
?	077	?	321	?	007	?	133	71
<	074	<	243	NULL	———	<	160	72
>	276	>	047	NULL	———	>	150	73
@	300	≤	044	@	020	≤	004	74
\	134	\	077	NULL	———	\	007	75
^	176	¯	042	NULL	———	¯	020	76
;	273	;	074	;	153	;	073	77
`	140	NULL	———	NULL	———	NULL	———	7600
a	341	α	101	a	171	α	171	7601
b	342	⊥	102	b	166	⊥	166	7602
c	143	∩	303	c	172	∩	172	7603
d	344	⌊	104	d	052	⌊	052	7604
e	145	∈	305	e	112	∈	112	7605
f	146	X	134	f	163	X	023	7606
g	347	∇	107	g	043	∇	043	7607
h	350	Δ	110	h	046	Δ	046	7610
i	151	ι	311	i	031	ι	031	7611
j	152	∘	312	j	103	∘	103	7612
k	353	¢	336	k	032	NULL	———	7613
l	154	□	314	l	106	□	106	7614
m	355	|	115	m	141	|	141	7615
n	356	T	116	n	122	T	122	7616
o	157	O	317	o	105	O	105	7617
p	360	←	100	p	013	←	001	7620
q	161	→	134	q	133	→	101	7621
r	162	ρ	322	r	051	ρ	051	7622
s	363	⌈	123	s	045	⌈	045	7623
t	164	~	324	t	002	~	002	7624
u	365	↓	125	u	062	↓	062	7625
v	366	U	126	v	061	U	061	7626
w	167	ω	327	w	165	ω	165	7627
x	170	⊃	330	x	142	⊃	142	7630
y	371	↑	131	y	147	↑	147	7631

3AE3A

† ON TTY MODELS HAVING NO UNDERLINE, THE BACKARROW (←) TAKES ITS PLACE

TABLE B-2. 64-CHARACTER SET (Cont'd)

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	
Z	372	⊂	132	Z	124	⊂	124	7632
{	173	{	140	NULL	———	NULL	———	7633
:	174	≥	246	±	040	≥	130	7634
}	175	}	374	NULL	———	NULL	———	7635
~	176	≡	175	NULL	———	NULL	———	7636
DEL	377	DEL	377	NULL	———	NULL	———	7637
NUL	000	NUL	000	NUL	075	NUL	075	7640
SOH	201	SOH	201	SOA	166	SOA	166	7641
STX	202	STX	202	EOA	064	EOA	064	7642
ETX	003	ETX	003	NULL	———	NULL	———	7643
EOT	204	EOT	204	EOT	174	EOT	174	7644
ENQ	005	ENQ	005	NULL	———	NULL	———	7645
ACK	006	ACK	006	ACK	067	NULL	———	7646
BELL	207	BELL	207	NULL	———	NULL	———	7647
BS	210	BS	210	BS	135	BS	135	7650
HT	011	HT	011	HT	057	HT	057	7651
LF	012	LF	012	LF	156	LF	156	7652
VT	213	VT	213	NULL	———	NULL	———	7653
FF	014	FF	014	NULL	———	NULL	———	7654
CR	215	CR	215	CR	155	CR	155	7655
SO	216	SO	216	UCS	034	UCS	034	7656
SI	017	SI	017	LCS	037	LCS	037	7657
DLE	220	DLE	220	NULL	———	NULL	———	7660
DC1	021	DC1	021	NULL	———	NULL	———	7661
DC2	022	DC2	022	NULL	———	NULL	———	7662
DC3	023	DC3	023	NULL	———	NULL	———	7663
DC4	024	DC4	024	STO	054	STO	064	7664
NAK	225	NAK	225	NAK	001	NAK	001	7665
SYN	226	SYN	226	IL	075	IL	075	7666
ETB	027	ETB	027	EOB	136	EOB	136	7667
CAN	030	CAN	030	DEL	177	DEL	137	7670
EM	231	EM	231	NULL	———	NULL	———	7671
SUB	232	SUB	232	NULL	———	NULL	———	7672
ESC	033	ESC	033	PF	076	PF	076	7673
FS	234	FS	234	NULL	———	NULL	———	7674
GS	035	GS	035	NULL	———	NULL	———	7675
RS	036	RS	036	NULL	———	NULL	———	7676
US	237	US	237	NULL	———	NULL	———	7677
NULL	———	NULL	———	NULL	———	NULL	———	7400
@	300	≤	044	@	020	≤	004	7401
^	176	—	042	NULL	———	—	020	7402
NULL	———	NULL	———	CNL	001	CNL	001	7403
NULL	———	NULL	———	NULL	———	NULL	———	7404
NULL	———	NULL	———	NULL	———	NULL	———	7405
NULL	———	NULL	———	NULL	———	NULL	———	7406
NULL	———	NULL	———	NULL	———	NULL	———	7407

3AE5A

PROGRAM CONTROL OF TERMINAL ACTIVITY C

The user can design an interactive program to control terminal activity in two ways.

- Include control bytes in his output to control the positioning of the printing element, define alternate input modes, etc.
- Issue a DISTC macro to disable the terminal operator's control of his program during critical phases of execution.

CONTROL BYTES

A control byte is a 12-bit quantity, right-justified in bit positions 0, 12, 24, 26, or 48 of a CM word. The user must be careful that data is not mistaken for a control byte. For example, the characters :D typed at the end of a line will log-off the user, since the code 0004 is transmitted. The following paragraphs describe the bytes available to the user and their functions.

END-OF-LINE

This byte generates a carriage return and line feed, positioning the terminal printing element at the beginning of the next line. An end-of-line consists of 12 to 66 bits of 0, right-justified in one or two CM words.

END-OF-BLOCK (0001 or 0002)

This byte prevents the positioning of the terminal printing element at the beginning of the next line. An end-of-block can be used to allow the terminal operator to enter input on the same line as the input request is printed. This byte must be followed by an end-of-line.

AUTO INPUT (0003)

This byte is intended for use by the time-sharing executive for auto mode input, † but the user may include it in his output. The preceding n characters in the word in which this byte occurs are also retained as the first n characters of the input line (n must be even and less than 9). This byte must be the first byte and must be followed by an end-of-line.

LOG-OFF USER (0004)

This byte disconnects the terminal user's telephone lines. This byte must be the first byte and must be followed by an end-of-line.

---

† Refer to the KRONOS Reference Manual, Volume 2 for further information about the DISTC macro.

SET TRANSPARENT INPUT MODE (0005)

This byte changes the input mode from normal or ASCII (refer to appendix F, volume 1) to transparent mode. A 0005 byte directs the driver to translate all characters to the 6/12 bit internal codes as in ASCII mode but to ignore no characters. Also, no input control characters except return and interrupt (delete, backspace, STEXT, and ETEXT) are processed. Rather, all characters are passed to the program as data. Interrupt is passed as null followed by end-of-line. Return is passed as end-of-line. End-of-line terminates transparent input mode. This byte must be the first byte and must be followed by an end-of-line.

SET BINARY INPUT MODE (0006)

This byte changes the input mode from normal or ASCII (refer to section 4) to binary mode. The control byte must be byte 0. Bytes 1 and 2 are defined as follows:

Byte 1	Specifies the maximum number (octal) of characters to be received before input is terminated. If byte 1 is 0, the number is assumed to be 1. Byte 1 cannot exceed 150.
Byte 2	Specifies the termination code. When a character is received from the terminal that matches the lower eight bits of this byte, the input operation is terminated. If bit 11 of this byte is set, no termination character is assumed.

This conversion mode packs the eight bits of data as the lower eight bits of a 12-bit byte and sets the upper bit (bit 11). The exhaustion of the character count or the occurrence of the termination code causes the end-of-line condition to be set. A 0007 byte is forced as the first byte of input so that data will be transmitted as binary if it is listed. This byte must be the first byte and byte 2 must be followed by an end-of-line byte.

INITIATE BINARY OUTPUT (0007)

This byte initiates binary output. If the user wishes to output data formatted as described for binary input, a 0007 byte must precede the data. This mode continues until another control byte is detected.

A control byte (0003, 0005, 0006) that changes the terminal input mode prevents the system from printing a question mark in response to a program request for input. However, in all other cases, a read request on the input buffer causes the system to print a question mark at the terminal.

END OF TRANSACTION BLOCK (0010)

This byte is used by TRANEX only to indicate the end of a transaction block. This byte must be terminated by an end-of-line byte.

INITIATE ASCII OUTPUT (0011)

This byte is used to initiate ASCII output. This byte must be byte 0 and must be issued for each ASCII line.

PAPER TAPE OPERATIONS D

Paper tape is used for preparing input off-line (when time is not charged) and entering it on-line (when time is charged). This ensures accuracy and speed when most needed. An input tape can include programs, data, and commands. Accordingly, it is possible for the entire terminal operation, after log-in, to be run from paper tape.

This exposition assumes a typical teletypewriter (model 33) with a paper tape punch and a paper tape reader.

The paper tape punch has four buttons with the following labels and use.

ON	Turns the punch on
OFF	Turns the punch off
REL.	Releases the feedwheel so one can freely move the tape through the punch head
B.SP.	Backspaces the tape one row of holes each time it is pressed. This is used to make corrections when preparing a tape off-line (refer to detailed description later in this appendix).

The paper tape reader has one switch with four positions. Their labels and use are as follows:

START	Momentarily pressing the switch to this uppermost position starts the reader. The switch snaps back to the AUTO position and reading continues.
STOP	Reading stops immediately whenever the switch is moved to this position.
AUTO	This position is used in conjunction with the input and processing of commands and data in tape mode (tape mode is explained later in this appendix). It allows the tape reader to be turned off and on so that processing of each command or line of data can be completed before additional input is entered.
FREE	Releases the feedwheel so one can freely slide the tape in and out of the read head.

NOTE

On teletypewriters lacking an AUTO position, reading must be manually restarted each time it has been stopped.

CONTROL CHARACTERS

Each line of input from paper tape must end with a particular sequence of control characters. These are punched by functional keys on the teletypewriter keyboard. The control characters used with paper tape are as follows.

CR	Carriage return. This is the RETURN key on the keyboard. It moves the print head back to the beginning of the line and informs the system that this line of input is completed.

LF	Line feed. This is the LINE FEED key on the keyboard. It advances the paper roller one line. In normal mode, the system sends back a signal that initiates this feed; in tape mode, this signal is not sent and is therefore required on tape.
CTRL/ X-OFF	X-OFF. The appearance of this character on the tape during reading turns the tape reader off. This character is punched by holding down the CTRL key and pressing the X-OFF key.
RO(n)	Rubout. This is the RUB OUT key on the keyboard. It punches a full row of holes. This row is interpreted as null by the system and hence this character is used for spacing and overpunching errors. The n parameter specifies the minimum number of times this character should be punched in sequence.
NUL	Null. This is the ASCII character represented by a feed hole only (blank tape). It may be used as a fill character.

INPUT LINES

The following line formats are used to enter programs, data, and commands. The left half of the page shows an example of the input and the right half shows the control characters that immediately follow the last input character. It should be noted that each line is terminated with three rubouts to provide separation from the next line. While it is possible that adequate separation may be provided with less than three rubouts, this is the recommended number. (The system punches three rubouts at the end of each line when a tape is produced on-line.)

PROGRAM LINE

```
100 PROGRAM TEST (INPUT,OUTPUT) -      - CR   LF   RO(3)
```

DATA LINE

```
? 12.44,18.31,29.08 -                  - CR   LF   CTRL/X-OFF   RO(3)
```

The ? is supplied by the system. The remainder of the line comes from the tape. The X-OFF turns the reader off to allow this data to be processed before the next line is read.

COMMAND LINE

```
CATLIST/LO=F -                         - CR   LF   CTRL/X-OFF   RO(3)
```

The X-OFF turns the reader off to allow processing of the command to be completed before the next line is read.

PUNCHING A PAPER TAPE OFF-LINE

The following procedure is used to punch a tape when the teletypewriter terminal is not logged-in.

1. Turn the teletypewriter on in local mode by pressing the LCL button below the phone dial.
2. Turn the paper tape punch on by pressing the ON button located on the punch.

3. Prepare a tape leader of about 30 rubouts (3 inches) by pressing the RUB OUT and repeat (REPT) keys togehter, or by punching blank tape if the terminal has the capability.
4. With a pencil, trace the arrow above the punch output onto the tape. This will identify the start of the tape for reading. (It is possible to put the tape in the reader backwards.)
5. Type in the input lines with their appropriate control characters.
6. Add a 3-inch trailer of rubouts or blank tape and tear the tape off.
7. Press the CLR button below the phone dial to turn the teletypewriter off.

TAPE MODE

To read and process data and commands from tape, it is necessary that the tape reader be turned off and on to allow time for processing. By entering tape mode, the user enables the system to synchronize tape input with its processing. This mode also inhibits the output of header messages and the READY message when LIST (or LNH) is used to punch a tape on-line (refer to Punching a Tape On-line). It should be noted that carriage return (CR) entered on an empty input line will be ignored by the system while in tape mode.

The user enters tape mode by typing the TAPE command and pressing the carriage return key. The system acknowledges entry into this mode by advancing the printout paper one line. It will not print READY. If the reader switch has been in the AUTO position, reading begins automatically; if the reader switch was in the OFF position, reading does not begin until manually initiated by momentarily moving the switch to START. All subsequent commands and data entries are acknowledged by the one-line advancement without the printout of READY.

The user exits from tape mode by typing the NORMAL command and pressing the carriage return key.

PUNCHING A TAPE ON-LINE

To punch a tape with information already contained in a system file, the following procedure is followed.

1. If not logged-in, log-in.
2. If not in tape mode, type TAPE and press the carriage return key.
3. Turn the punch on.
4. Run a leader of rubouts or blank tape.
5. Type in LIST if the primary file is to be punched; type in LIST,F=lfn if a working file other than the primary file is to be punched. (The short form LNH is permissible.)
6. Press the carriage return key. The file will be listed and punched simultaneously. The system will add the appropriate control characters at the end of each line.
7. Run a trailer of rubouts or blank tape.

CORRECTIONS

While punching a tape off-line, corrections can be made by backspacing over the incorrect punch (use the B.SP. button on the punch) and punching a rubout over the error (use the RUB OUT key on the keyboard). Then punch in the correct character.

It is also possible to make corrections by using the back arrow as in normal, on-line operation. However, the error and arrow will appear in the printout. Likewise, when line numbers are being used, an erroneous line can be retyped (and repunched) with the same line number. Again, the erroneous and correct version will appear in printout although the system will recognize only the correct version during processing.

Example:

In this example, the input tape contains only program lines. Commands and input are typed in by the user before and after the tape is read in.

The tape as punched is shown below. The left half of the page gives the printed copy which was produced as the tape was punched; the right half of the page shows the control characters which were entered at the end of each line but were not printed. In this example, any character shown as RO (rub-out) may be replaced by NUL (blank tape).

```
                                    RO(30) or NUL(30)
100 LET FACTOR=1                    CR   LF   RO(3)
110 INPUT N                         CR   LF   RO(3)
120 FOR I=1 TO N                    CR   LF   RO(3)
130 LET FACTOR=FACTOR*I             CR   LF   RO(3)
140 PRINT FACTOR                    CR   LF   RO(3)
150 NEXT I                          CR   LF   RO(3)
160 END                             CR   LF   RO(30)
```

To demonstrate the use of this tape, it is assumed the user is logged-in under the NULL subsystem. He positions this tape in the reader and sets the reader switch to AUTO. The following is the printout that results from the interaction of user and tape with the terminal. The right half of the page contains explanations of the action.

```
BASIC  <----------------------------   By typing in this command, the user
READY.                                 enters the BASIC subsystem.

NEW,FACT1  <------------------------   This line is typed in to establish a
READY.                                 new primary file with the name FACT1

TAPE  <-----------------------------   The command TAPE is typed in, and as
                                       soon as the carriage return takes effect,
                                       the tape reader goes on and the entire
100 LET FACTOR=1                       tape (seven lines) is read in.
110 INPUT N                            The reader runs through the trailer of
120 FOR I=1 TO N                       rubouts and then turns itself off.
130 LET FACTOR=FACTOR*I
140 PRINT FACTOR
150 NEXT I
160 END
RNH  <------------------------------   Types in RNH and receives the diagnostics
  ILLEGAL STATEMENT AT 100             shown. BASIC variables cannot be more
  ILLEGAL STATEMENT AT 130             than two characters.
  ILLEGAL STATEMENT AT 140
```

```
SRU          0.015 UNTS.

RUN COMPLETE.

100 LET F=1
130 LET F=F*I
140 PRINT F
RNH <-------------------------------------
                                            Initiates another run.
                                            The system replies with ?
? 5                                         He types in 5 and receives the
                                            desired printout.
1
2
6
24
120

SRU          0.017 UNTS.

RUN COMPLETE.

LIST <------------------------------------
100 LET F=1                                 The user wants to punch a new tape con-
110 INPUT N                                 taining this corrected program. He types
120 FOR I=1 TO N                            in LIST but does not press the carriage
130 LET F=F*I                               return key. He turns on the punch, runs
140 PRINT F                                 off some leader, and then enters the
150 NEXT I                                  carriage return. As the system prints the
160 END                                     listing, it also punches a tape with the
                                            same information along with the control
                                            characters CR, LF, and RO(3) at the end
                                            of each line. After this, the user punches
                                            a trailer of rubouts and tears off the
                                            new tape. He discards the original tape.
```

Example:

In this example, the input tape contains not only a program but commands to execute, list, modify, and save that program, as well as the input data used by that program.

The complete tape as punched is shown below. The left half of the page gives the printed copy which was produced as the tape was punched; the right half of the page shows the control characters which were entered at the end of each line but were not printed. In this example, any character shown as RO (rub-out) may be replaced by NUL (blank tape).

				RO(30) or NUL(30)
FORTRAN	CR	LF	CTRL/X-OFF	RO(3)
NEW,DEMO	CR	LF	CTRL/X-OFF	RO(3)
100 PROGRAM DEMO(INPUT,OUTPUT)	CR	LF	RO(3)	
110 DIMENSION II(5)	CR	LF	RO(3)	
120 READ 10, (II(J),J=1,5)	CR	LF	RO(3)	
130 10 FORMAT(5I5)	CR	LF	RO(3)	
140 PRINT 20, (II(J),J=1,5)	CR	LF	RO(3)	
150 20 FORMAT(5I10)	CR	LF	RO(3)	
160 END	CR	LF	RO(3)	
RNH	CR	LF	CTRL/X-OFF	RO(3)
1111122222333334444455555	CR	LF	CTRL/X-OFF	RO(3)
SAVE,DEMO=TAPE1	CR	LF	CTRL/X-OFF	RO(3)
135 DO 1 I=1,5	CR	LF	RO(3)	
137 II(I)=II(I)+4444	CR	LF	RO(3)	
139 1 CONTINUE	CR	LF	RO(3)	
RESEQ,100,5	CR	LF	CTRL/X-OFF	RO(3)
LIST	CR	LF	CTRL/X-OFF	RO(3)
RNH	CR	LF	CTRL/X-OFF	RO(3)
1111122222333334444455555	CR	LF	CTRL/X-OFF	RO(3)
REPLACE,DEMO=TAPE1	CR	LF	CTRL/X-OFF	RO(30)

To execute this tape, the user logs in, positions the tape in the reader, sets the reader switch to AUTO, types in TAPE, and upon pressing the carriage return key, starts the reading of tape. He then observes the remainder of the action without intervention.

The following is the printout of the execution of the tape. The right half of the page contains explanations of the action.

```
FORTRAN
```
When this line is read, the reader stops. As soon as FORTRAN is established as the current subsystem, the system turns the reader back on.

```
NEW,DEMO
```
When this line is read, the reader stops. After a new primary file called DEMO is established, the system turns the reader back on.

```
100 PROGRAM DEMO(INPUT,OUTPUT)
110 DIMENSION II(5)
120 READ 10,(II(J),J=1,5)
130 10 FORMAT(5I5)
140 PRINT 20,(II(J),J=1,5)
150 20 FORMAT(5I10)
160 END
```
The seven lines of the program are read in without interruption.

```
RNH
```
The command to run the program is read in and the reader turned off.

```
? 111112222233333444445555
```
Processing reaches the READ statement (line 120) and the system prints ?. The system then turns the reader back on and the line of data is read in. Then the reader is turned off to allow data processing.

Resultant printout

```
    11111     22222     33333     44444     55555
END.


SRU          0.130 UNTS.

RUN COMPLETE.
```

```
SAVE,DEMO=TAPE1
```

This command is read in and the reader turned off. A copy of the program demo is made an indirect access permanent file with the name TAPE1. Then the reader is turned back on.

```
135 DO 1 I=1,5
137 II(I)=II(I)+4444
139 1 CONTINUE
```

These three lines of modification are read in from the tape without interruption.

```
RESEQ,100,5
```

The RESEQ command is read in and the reader turned off. When resequencing is accomplished according to specifications, the reader is turned back on.

```
SRU           0.019 UNTS.
LIST
```

This command is read from tape and the tape reader turned off. Then the system lists the primary file which contains the above modifications and resequenced format. When the listing is completed, the reader is turned on again.

```
00100 PROGRAM DEMO(INPUT,OUTPUT)
00105 DIMENSION II(5)
00110 READ 10,(II(J),J=1,5)
00115 10 FORMAT(5I5)
00120 DO 1 I=1,5
00125 II(I)=II(I)+4444
00130 1 CONTINUE
00135 PRINT 20,(II(J),J=1,5)
00140 20 FORMAT(5I10)
00145 END

RNH
```

This command is read in, the reader turned off, and the modified program executed. The system prints the ? and then turns the reader on. The line of data is read in from the tape and the reader turned off. The data is processed and results printed.

```
? 111112222233333444445555

    15555      26666      37777      48888      59999
END.

SRU           0.141 UNTS.

RUN COMPLETE.

REPLACE,DEMO=TAPE1
```

This last command is read in and the reader turned off. A copy of this revised version of DEMO replaces the old one that was made an indirect access permanent file under the name TAPE1. It is now the indirect access permanent file that is referenced by the name TAPE1.

The reader is turned back on. It runs through the trailer of rubouts and then turns itself off.

If, at this point, the user has no more tapes to run, he should exit tape mode by issuing the NORMAL command.

MASS STORAGE DEVICE STATISTICS E

The following list describes the mass storage devices supported by KRONOS and the size of blocks allocated for direct access files.

Device Code	Device	PRUs	Block Size	
			CM Words	Characters
DE	Extended Core Storage	16	1024	10240
DIn	844 Disk Storage Subsystem ($1 \leq n \leq 8$)	n*107	n*6848	n*68480
DP	Distributive Data Path	16	1024	10240
MDn	841 Multiple Disk Drive ($1 \leq n \leq 8$)	n*32	n*2048	n*20480

In this table, n indicates the unit count for multiunit devices.

If the user's validation permits (refer to LIMITS command, section 4), the maximum size indirect access file is determined by the device limit less the space allocated for catalog information and other files.

The size of a direct access file is limited only by the device; there are no system limitations.

TEXT EDITOR COMMAND INDEX F

This is a list of the commands available under Text Editor. For additional information, refer to the Text Editor (EDIT) Reference Manual.

ADD(S)
ADD(S);n
ADD(S):/string/
ADD(S):/string/;n

ALIGN
ALIGN;n
ALIGN:/string/
ALIGN:/string/;n
ALIGN:/string1/, /string2/
ALIGN:/string1/, /string2/;n

BLANK(S)
BLANK(S);n
BLANK(S):/string/
BLANK(S):/string/;n
BLANK(S):/string1/, /string2/
BLANK(S):/string1/, /string2/;n

CHANGE(S)
CHANGE(S);n
CHANGE(S):/string/
CHANGE(S):/string/;n
CHANGE(S):/string1/, /string2/
CHANGE(S):/string1/, /string2/;n

CLEAR

DEFTAB
DEFTAB:/tabchar/

DELETE(S)
DELETE(S);n
DELETE(S):/string/
DELETE(S):/string/;n
DELETE(S):/string1/, /string2/
DELETE(S):/string1/, /string2/;n

END

EXTRACT
EXTRACT;n
EXTRACT:/string/
EXTRACT:/string1/, /string2/
EXTRACT:/string1/, /string2/;n

ES
ES;n
ES:/string/
ES:/string/;n
ES:/string1/, /string2/
ES:/string1/, /string2/;n

FIND(S)
FIND(S);n
FIND(S):/string/;n
FIND(S):/string1/, /string2/;n

INSERTS:/string1/, /string2/;n

LENGTH;n.
LENGTH;*

LINE

LIST(S)
LIST(S);n
LIST(S):/string/
LIST(S):/string/;n
LIST(S):/string1/, /string2/
LIST(S):/string1/, /string2/;n

LISTAB

MERGE:/lfn/;n
MERGE:/lfn/, /string/;n

NUMBER(S)
NUMBER(S):/string/
NUMBER(S):/string1/, /string2/

RS:/string/;n
RS:/string1/, /string2/;n

RESET

SET
SET, n
SET-n
SET:/string/
SET:/string/;n

TAB
TAB:/$t_1, \ldots, t_n$/

WIDTH;n ($6 \leq n \leq 150$)

RESERVED FILE NAMES G

Several file names are reserved for system use or have special significance to the system. These reserved names protect the user from accidentally destroying some of his own files. The message RESERVED FILE NAME. indicates that the user has attempted to use one of these file names.

The reserved file names are:

INPUT
OUTPUT
PUNCH
PUNCHB
P8
SCR
SCR1
SCR2
SCR3
SCR4

An attempt is not always made to prevent the use of these file names. Many of the product set members, such as the COMPASS assembler, use other scratch files in addition to those listed. A user should consult the appropriate reference manuals for a complete list of these files. System integrity does not depend on preventing the use of these reserved file names.

SAMPLE SESSION AT THE TERMINAL H

This appendix presents a typical interaction of a beginning user with the KRONOS Time-Sharing System. After log-in, the user tries basic operations in the various subsystems and tests a cross section of the commands available.

The left half of the page is the terminal (Teletype model 33) printout of the interaction. The right half of the page gives explanations. In a few places the terminal printout has been vertically expanded to allow spacing for the inclusion of the relevant comments.

```
 75/12/28.  10.34.14.
KRONOS TIME SHARING SYSTEM - VER. 2.1.2.
USER NUMBER: USER123
PASSWORD
■■■■■■■■■
TERMINAL:    40,  TTY
RECOVER/SYSTEM: NULL
OLD, NEW, OR LIB FILE:  NEW,TEST
READY.

TEXT
ENTER TEXT MODE
LINE 1: ABCDEFGHIJ
LINE 2:   BCDEFGHIJK
LINE 3:     CDEFGHIJKL
LINE 4:       DEFGHIJKLM
LINE 5:         EFGHIJKLMN
LINE 6:           FGHIJKLMNO
LINE 7:             GHIJKLMNOP
LINE 8:               HIJKLMNOPQ
LINE 9:                 IJKLMNOPQR
LINE 10:                 JKLMNOPQRS

EXIT TEXT MODE.
```

The user intends to create a primary file that he will use for experimental purposes. He will preserve this file as an indirect access permanent file. However, he does not want a particular subsystem associated with the file. Accordingly, he enters the NULL subsystem.

User enters text mode because the source data to be entered will not have line numbers.

Types in an array of alphanumeric characters to be used in testing the Text Editor.

User exits text mode by pressing the BREAK key.

```
LIST,R
LINE 1: ABCDEFGHIJ
LINE 2:  BCDEFGHIJK
LINE 3:   CDEFGHIJKL
LINE 4:    DEFGHIJKLM
LINE 5:     EFGHIJKLMN
LINE 6:      FGHIJKL--EOR--
MNO
LINE 7:       GHIJKLMNOP
LINE 8:        HIJKLMNOPQ
LINE 9:         IJKLMNOPQR
LINE 10:         JKLMNOPQRS
--EOR--
READY.
```

Remembers that Text Editor can only operate on a single record. He suspects he has created more than one record and calls for a listing with end-of-record marks.

His suspicions are confirmed.

```
PACK
READY.
```

The PACK command compresses the file into one record.

```
LNH,R.
LINE 1: ABCDEFGHIJ
LINE 2:  BCDEFGHIJK
LINE 3:   CDEFGHIJKL
LINE 4:    DEFGHIJKLM
LINE 5:     EFGHIJKLMN
LINE 6:      FGHIJKLMNO
LINE 7:       GHIJKLMNOP
LINE 8:        HIJKLMNOPQ
LINE 9:         IJKLMNOPQR
LINE 10:         JKLMNOPQRS
--EOR--
READY.
```

This is verified with another listing.

```
SAVE,TEST=TEXT1
READY.
```

In addition to using this file in the subsequent EDIT operations, the user feels he may need it for other testing and makes a permanent file (indirect access) called TEXT1. The original copy, called TEST, remains as the primary file.

```
EDIT
```

Enters the Text Editor

```
 BEGIN TEXT EDITING.
? LIST

LINE 1: ABCDEFGHIJ
```

List the first line. Pointer remains at the first line.

```
? LIST;3

LINE 1: ABCDEFGHIJ
LINE 2:  BCDEFGHIJK
LINE 3:   CDEFGHIJKL
```

List the first three lines.

```
? LIST:/JKL/

LINE 3:   CDEFGHIJKL
```

List the first line in which the phrase /JKL/ occurs

```
? LIST:/JKL/;3
```
List the first three lines in which the phrase /JKL/ occurs.

```
LINE 3:     CDEFGHIJKL
LINE 4:      DEFGHIJKLM
LINE 5:       EFGHIJKLMN
? LIST:/JKL/,/GHI/
```
List the line(s) in which the string beginning with /JKL/ and ending with /GHI/ first occurs.

```
LINE 3:     CDEFGHIJKL
LINE 4:      DEFGHIJKLM
? LIST:/JKL/,/GHI/;3
LINE 3:     CDEFGHIJKL
LINE 4:      DEFGHIJKLM
LINE 5:       EFGHIJKLMN
LINE 6:        FGHIJKLMNO
? LISTS:/DEFG/
```
List the lines that contain the first three occurrences of the string beginning with /JKL/ and ending with /GHI/.

List the first occurrence of the phrase /DEFG/.

```
            DEFG
? LISTS:/DEFG/;3
```
List the first three occurrences of the phrase /DEFG/.

```
            DEFG
            DEFG
            DEFG
? LISTS:/LMN/,/NOP/
                    LMN
LINE 6:        FGHIJKLMNO
LINE 7:         GHIJKLMNOP
? FIND;5
LINE 6:        FGHIJKLMNO
```
List the first occurrence of the string beginning with /LMN/ and ending with /NOP/.

Move the pointer five lines and list the new line. Pointer remains at new line.

```
? FIND:/DEF/;3
PHRASE NOT FOUND.
```
Move the pointer another three lines and list the first line in which the phrase /DEF/ occurs. Pointer has moved beyond the range of this phrase and hence it is not found.

```
?RESET
```
Moves the pointer back to the beginning of the file.

```
? FINDS:/DEF/;3
            DEF
```
Moves the pointer three lines and lists the first occurrence of /DEF/.

```
? RESET
```
Moves the pointer back to the first line.

```
? ADD;2
```
Calls for text to be added after the second line.

```
 ENTER TEXT.
? /FIRST INSERTION/
 READY.
? ADD;4 <-------------------------- Calls for text to be entered after
 ENTER TEXT.                        the fourth line.
? /SECOND INSERTION/
 READY.
? ADD:/MNO/ <---------------------- Calls for text to be entered after the
 ENTER TEXT.                        first line in which /MNO/ occurs.
? /THIRD INSERTION/
 READY.
? ADDS:/MNO/ <--------------------- Calls for text to be entered immediately
 ENTER TEXT.                        after the phrase /MNO/.
? /FOURTH INSERTION/
 READY.
? SET;8 <-------------------------- The pointer is moved down eight lines
? DELETE;2                          and then the next two lines are deleted.
? RESET                             Taking into account the above insertions,
?LIST;*                             it should stop at /THIRD INSERTION/ and
LINE 1: ABCDEFGHIJ                  then eliminate that line and the next
LINE 2:  BCDEFGHIJK                 one (line 7).
FIRST INSERTION
LINE 3:   CDEFGHIJKL                The pointer is moved back to start and
SECOND INSERTION                    a listing made of the entire file.
LINE 4:    DEFGHIJKLM
LINE 5:     EFGHIJKLMN
LINE 6:      FGHIJKLMNOFOURTH INSERTION
LINE 8:        HIJKLMNOPQ
LINE 9:         IJKLMNOPQR
LINE 10:         JKLMNOPQRS
 -END OF FILE-
? OLD,TEXT1
OLD         SYNTAX ERROR.           The user would like to get back the
? END                               original format of this file.  Since the
 END TEXT EDITING.                  original is an indirect access permanent
READY.                              file, he calls for a copy to be entered
                                    as the primary file (OLD).  But commands
OLD,TEXT1                           cannot be entered from the Text Editor.
READY.                              Accordingly, he exits from EDIT, enters
                                    his OLD, and then reenters EDIT.
EDIT
 BEGIN TEXT EDITING.
? CHANGE;3 <----------------------- Change the first three lines to the
 ENTER TEXT.                        text that follows.
? /AAA
? BBB
? CCC/
 READY.
? CHANGE:/JKLM/ <------------------ Change the first line with the phrase
 ENTER TEXT.                        /JKLM/ to the text that follows.
? /OO
? OO
? OO/
 READY.
```

```
? LIST;*
AAA
BBB
CCC
OO
OO
OO
LINE 5:       EFGHIJKLMN
LINE 6:        FGHIJKLMNO
LINE 7:         GHIJKLMNOP
LINE 8:          HIJKLMNOPQ
LINE 9:           IJKLMNOPQR
LINE 10:           JKLMNOPQRS
 -END OF FILE-
? END
 END TEXT EDITING.
READY.
```

A listing of the modified file verifies the changes.

Exits from the Text Editor.

```
BASIC
READY.
```

Enters the BASIC subsystem.

```
NEW,DEMO
READY.
```

Establishes new primary file.

```
AUTO
00100 INPUT N
00110 PRINT TAB(169;"INTEGERS","SUM"
00120 LET S=0
00130 FOR I=1 TO N
00140 S=S+I
00150 PRINT TAB(19);I,S
00160 NEXT I
00170 END
00180  *DEL*
NORMAL
READY.
```

Enters AUTO mode so that the system will supply line numbers

Deletes this line so he can type NORMAL and exit AUTO mode.

```
90 REM SUMMATION OF THE FIRST N INTEGERS
95 REM THE VALUE OF N TO BE USSP____ SUPPLIED AT EXECUTION
```

Adds comments to beginning of program.

```
RESEQ,1000,1000
```

Resequences line numbering to begin at 1000 and increase in increments of 1000.

```
SRU          0.028 UNTS.
READY.
```

```
LIST                                          A listing shows the new line numbers

 75/12/29.  10.46.27.
PROGRAM  DEMO

01000 REM SUMMATION OF THE FIRST N INTEGERS
02000 REM THE VALUE OF N TO BE SUPPLIED AT EXECUTION
03000 INPUT N
04000 PRINT TAB(169;"INTEGERS","SUM"
05000 LET S=0
06000 FOR I=1 TO N
07000 S=S+I
08000 PRINT TAB(19);I,S
09000 NEXT I
10000 END
READY.

RNH

 ILLEGAL STATEMENT AT  4000 <------Refers to typo error in line 04000.


SRU          0.017 UNTS.

RUN COMPLETE.

04000 PRINT TAB(16);"INTEGERS","SUM" Corrected statement.
RNH

? 10                                          Program is executed and the input
                                              for N is 10.
                                              The printout follows.
                 INTEGERS      SUM
                     1           1
                     2           3
                     3           6
                     4          10
                     5          15
                     6          21
                     7          28
                     8          36
                     9          45
                    10          55

SRU          0.037 UNTS.

RUN COMPLETE.
```

```
SAVE, DEMO=DEMO1
ILLEGAL PARAMETER.

SAVE,DEMO=DEMO1
READY.
```

Decides to make this program an indirect access permanent file called DEMO1. The first attempt fails because of the space after the comma. Note DEMO1 will have a BASIC subsystem flag associated with it because that subsystem is in effect when the file is saved. Hereafter, when the file is accessed (using the OLD command), the BASIC subsystem will also be selected.

```
1000 REM SUMMATION OF THE ODD INTEGERS
2000
6000 FOR I=1 TO N STEP 2
RESEQ, 1000, 1000
```

Alters the program to sum the odd integers

```
SRU          0.028 UNTS.
READY.

RNH

? 8
```

Enters the value 8 for N in the request for input.

```
                INTEGERS      SUM
                    1           1
                    3           4
                    5           9
                    7          16

SRU          0.026 UNTS.

RUN COMPLETE.

SAVE,DEMO=DEMO2
READY.
```

Makes this altered program an indirect access permanent file which he calls DEMO2. This file will also have a BASIC subsystem flag associated with it.

```
NEW,TEST
READY.
```

Establishes a new primary file.

```
BATCH
```

Enters the BATCH subsystem.

```
$RFL,20000.
```

By default, the system supplies a field length of 20000 octal words.

```
/DEFINE,ALPHA
```

Creates an empty direct access permanent file called ALPHA.

```
/COPYCR(INPUT,ALPHA,3)
? FILE ALPHA - RECORD 1 - LINE 1
?                         LINE 2
?                         LINE 3
?
?              RECORD 2 - LINE 1
?                         LINE 2
?
?              RECORD 3 - LINE 1
?                         LINE 2
?                         LINE 3
?
COPY COMPLETE.
/LIST,F=ALPHA

COPY COMPLETE.
/REWIND,ALPHA

$REWIND,ALPHA.
/LIST,F=ALPHA

FILE ALPHA - RECORD 1 - LINE 1
                        LINE 2
                        LINE 3
             RECORD 2 - LINE 1
                        LINE 2
             RECORD 3 - LINE 1
                        LINE 2
                        LINE 3
COPY COMPLETE.

/FTNTS
READY.

NEW,TEST
```

Calls for input to the permanent file in the form of three records

This data is written directly on the direct access file without any local copy.

Calls for a listing of this direct access file.

Gets a null record because he has not rewound.

He rewinds and lists successfully.

Enters the FORTRAN Extended subsystem

Creates a program that generates tables of logarithms for various bases.

```
100CPROGRAM TO CREATE A TABLE OF LOGARITHMS
110CFOR ANY BASE FROM 0.1 TO 9.9
120 PROGRAM TEST(INPUT,OUTPUT)
130 READ, N, BASE
140 DO 1 I=1,N
150 READ, VALUE
160 BLOG - ALOG10(VALUE)/ALOG10(BASE)
170 PRINT 10, BASE, VALUE, BLOG
180 1 CONTINUE
190 10 FORMAT(9HLOG(BASE ,F3.1,2H ),F6.2,
200+3H = ,F10.8)
210 END
```

```
RNH

? 4,2 <----------------------------------- Input states that there will be four
? 2                                        entries for which the program will
LOG(BASE 2.0 )  2.00 = 1.00000000          determine the logarithms to the base two.
? 4
LOG(BASE 2.0 )  4.00 = 2.00000000
? 8
LOG(BASE 2.0 )  8.00 = 3.00000000
? 64
LOG(BASE 2.0 ) 64.00 = 6.00000000
 END.

SRU          0.157 UNTS.

RUN COMPLETE.

SAVE,TEXT=LOG1 <-------------------------- The primary file is made an indirect
READY.                                     access permanent file with the name LOG1.
                                           This file will have a FTNTS subsystem
                                           flag associated with it.

170 WRITE 10, BASE, VALUE, BLOG            He wants to modify the local copy of the
RNH                                        program so that it will write a local
                                           file rather than give a printout.

AT LINE NUMBER  170 UNRECOGNIZABLE STATEMENT
000001    FORTRAN ERRORS IN    TEST
                                           He gets the format wrong.

SRU          0.085 UNTS.

RUN COMPLETE.

120 PROGRAM TEST(INPUT, TAPE1)             He realizes that in addition to correcting
170 WRITE(1,10) BASE, VALUE, BLOG          statement 170, he must alter the header
RNH                                        line to establish a local file to write
                                           on.  He drops the OUTPUT file since
                                           there will be no printout.

? 3,8 <----------------------------------- Input states that there will be three
? 8                                        values entered for which the program is
? 64                                       to determine the logarithms to the base
? 512                                      eight.
 END.

SRU          0.164 UNTS.

RUN COMPLETE.
```

```
REWIND 1
ILLEGAL COMMAND.
```

He wants to see what he has written on the local file. He remembers to rewind but gets the form wrong.

```
REWIND,TAPE1
```

This is right.

```
SRU          0.001 UNTS.
READY.
```

```
LNH,F=TAPE1
```

Listing gives him what he wanted.

```
LOG(BASE 8.0 )  8.00 = 1.00000000
LOG(BASE 8.0 ) 64.00 = 2.00000000
LOG(BASE 8.0 )512.00 = 3.00000000


SRU          0.004 UNTS.
READY.
```

```
SAVE,TEST=LOG2
READY.
```

He makes an indirect access permanent file of this altered program which he calls LOG2. This file will also have an FTNTS subsystem flag associated with it.

```
BYE
```

The user logs off the system.

```
USER123     LOG OFF.    11.10.30.
USER123     SRU         0.870 UNTS.
```

INDEX

Abbreviating commands 4-1
Access mode 5-1
Accidental disconnect 4-16, 19; 7-5
ACCOUNT card image 7-2; 8-1
Accumulated CPU time 4-20
Acoustic coupler B-1
Alter line numbers 4-3
Alternate catalog requests 5-1, 3, 13
Alternate user access 5-1, 3, 18
APL character set B-5, 6
APL command 4-10
APL print selection (refer to TERM command)
APPEND command 5-6
Append mode 5-3
Answerback 4-23
ASCII character set 4-2, 3; B-5, 6
ASCII code terminals 1-2; 2-1, 2; 4-2, 3, 5, 9, 12, 19, 20, 22; 7-3; B-3
ASCII command 4-2, 3; 6-1
ATTACH command 5-1, 2, 6, 10, 22
Automatic generation of line numbers 3-3; 4-3
Automatic read-only permission 5-1
Automatic subsystem selection 3-3; 5-12
AUTO command 3-3; 4-3
Auto mode 4-3
Automatic rewind 3-3
Auxiliary device 5-1, 5, 6, 10

Backspace key 4-2, 3; 6-2; B-3
BASIC command 4-7
BASIC subsystem 1-2; 2-2; 7-2
BATCH command 4-8
Batch job image 8-1
Batch queue 4-21; 8-1, 5
BATCH subsystem 1-2; 4-7; 9-1
Baud rate switch 2-1
BINARY command 4-11
Binary file 4-8, 17
Binary input mode 4-10
Block size
  indirect access file 5-2
  direct access file D-1
BOI (beginning-of-information) 3-1; 8-5
BYE command 2-7; 3-7; 4-11; 5-7

CALL procedure file 4-24
Catalog entries 5-12, 13, 14, 15
CATLIST command 5-12, 16, 17, 18, 19, 20, 21, 22
Carriage return (refer to RETURN key)
c_1ECc_2 directive 8-5
Central memory character equivalents B-4
cEOF directive 8-3
cEOR directive 8-3
CHANGE command 5-7
Change working file name 4-16
Character count delay (refer to ROUT command)
Character set 4-2; B-3, 4
CHARGE command 2-3; 4-12
Charge/project number 2-3; 4-11; 6-8
cJOB directive 8-2
CLEAR command 3-7; 4-12; 5-7
cNOPACK directive 8-3, 5
cNOSEQ directive 8-3
cNOTRANS directive 8-2, 4
Command
  abbreviation 4-1
  entry of 4-1, 2
  error correction 4-2
  format, permanent files 5-2, 12
  index to (refer to inside front cover)
  minimum number of characters 4-1
  processing 4-1
  uniqueness 4-1
Compiling (refer to object code generation)
Connect terminal via acoustic coupler B-1
Connect terminal via data set B-1
Continue job after suspension 7-3
Control cards 4-22; 8-1; 9-1
Control keys B-3
Control language 4-22
Control record 8-5
Conventions, operating 1-2
CONVERT command 4-12
COR (refer to TERM command)
CORAPL (refer to TERM command)
Correcting errors 4-2
Correct input line 6-2
Corrections to paper tape C-4
Correcting statements in program 7-1
Correspondence code terminals 1-2; 2-1, 2; 4-2, 3, 5, 9, 10, 12, 19, 20, 22, 24; 7-3; B-3
cPACK directive 8-3, 5
CPU time, accumulated 4-20
cREAD directive 8-4

cREWIND directive 8-4,5
cSEQ directive 8-3
CSET command 4-4
cTRANS directive 8-2,3
CTRL key C-2

Delay carriage return (refer to ROUT command)
DEFINE command 5-1,5,8
Data set B-1
DAYFILE command 4-13
DAYFILE control card image 8-5
Delete input lines 4-3; 6-3
Delete program statement 7-1
Device availability 5-6,15
Dial-in procedure B-1
Direct access files 5-1
Disconnect terminal 2-5; 4-10,11
Display code B-5,6

EDIT command 3-5; 4-13
ENQUIRE command 4-14
Entry
  charge/project number 2-3; 4-11
  errors 4-2
  family name 6-5
  file name 2-3; 3-2,6; 4-6,9; 5-2,6; 6-8; 8-1
  file status 2-3; 4-6,20; 8-1
  lowercase letters 4-2,3
  numbers 4-2
  password 2-2; 4-16; 6-5
  subsystem 2-2,3; 4-6
  system commands 4-1
  terminal identification 2-2
  terminal type 4-5
  user number 2-2; 6-5
EOF (end-of-file) 3-1; 8-3,4,5
EOI (end-of-information) 3-1
EOR (end-of-record) 3-1; 8-3,4,5
Error correction 4-2
Error messages A-1,2,3
ESC (ESCAPE) key 4-2,3
Escape character 8-1
Escape code convention 4-3
Even/odd parity switch 2-1
Even parity 4-5
Examples
  AUTO command 4-4
  BASIC usage 2-4
  CATLIST command 5-16,17,18,19,20,21
  commands and effect on files 3-6
  delete input line 6-3
ESC (ESCAPE) key usage 4-4; 6-3
file modification 3-5; 7-1
listing a file 3-4
log-in 2-3; 6-9
ND parameter 3-6
PACKNAM command 5-11
paper tape usage C-4
program editing 7-1
program entry/execution 2-4
recovery 7-7,8,9
RUN command 4-18
Text Editor usage G-1
time limit 7-5
  time-sharing session G-1
  use of I key 7-4
  user validation limits 4-25
EXECUTE command 4-8
EXECUTE subsystem 1-2; 4-7
Execution of program 2-4; 3-6; 4-7,8,17,19; 6-10
EXIT control card image 8-5,6
Exit from BATCH subsystem 9-2
Export/Import 4-21; 8-1,5

Family name 2-3
Field length required for control cards 4-7; 9-1
Field length specified on RUN command 4-17
File
  catalog 5-1
  category 5-3
  direct access 5-1,2
  indirect access 3-2; 5-1
  inhibit sorting 3-4
  name entry 2-3; 3-2,6; 4-6,9; 5-2,6; 8-1
  organization 3-1
  permission information 5-13
  positioning 3-3; 4-17
  processing 3-1
  reformatting 8-2
  residency 5-14,14
  space specification 5-5
  sorting 3-3; 4-19
  status entry 2-3; 4-6,40; 8-1
  structure 3-1
  subsystem association 3-3
  usage 3-1
  working 3-2
  working copy of 5-1,2
FORTRAN subsystem 1-2; 4-8; 7-2
FTNTS command 4-9
FULL command 4-5
Full duplex/half duplex (FDX/HDX) switch 2-1; 4-4; 6-4

Generate binary (object code) file 4-17; 6-11
GET command 3-2; 5-9, 10, 22; 9-2
GOODBYE command 3-7; 4-16; 5-7
Guide to first session 2-1

HALF command 4-5
HELLO command 3-7; 4-16; 5-7
HELP command 4-12

I key 4-16; 7-3
Increment job field length 4-17
Indirect access files 3-2; 5-1
Inhibiting header messages 4-5
Input to executing program 6-3
Input/output conventions 6-1
Inserting statements in program 7-1
Interrupt program 4-12, 19; 6-3; 7-3
Introduction to KRONOS 1-1; 2-1
Invalid commands 4-2

JOB card image 8-2
Job
    name 4-21; 8-2
    status 4-20
    suspension 4-12; 7-3
    termination 4-19, 21

Keyboard configuration B-2
Keyword 5-2, 3, 12
Kilo physical records 4-13

LENGTH command 4-13, 16
Length of input/output lines 6-2
-lfn (procedure file call) 4-22
LIBRARY command 3-2, 6; 4-17; 5-1, 7, 10, 22
LIMITS command 4-17, 25
Line feed (LF) character C-2
Line numbers 3-3; 4-3
Line overflow 6-4
LIST command 3-4; 4-18
LNH (list-no-header) command 4-18
Local files (refer to working files)
Locked file 4-13
Logical record 3-1
LOGIN command 3-7; 4-18; 5-7
Log-in procedure 2-1
Log-off procedure 2-7
Lost data 6-4
Lost files 7-6

MEMAPL (refer to TERM command)
Minimum number of characters in command 4-1
Modify mode 5-3
Multifile file 3-1
Multiple file requests 9-2

ND parameter 3-6; 4-19; 5-10
NEW command 3-2, 6; 4-18; 5-6
No drop (refer to ND parameter)
NORMAL command 4-5; 6-1
Normal mode 4-5
NOSORT command 3-4; 4-18
NULL command 4-9
NULL subsystem 3-3; 4-9, 10; 5-10, 12

Object code execution 4-7, 8
Object code generation 4-17; 6-11
Odd parity 4-5
OLD command 3-2, 6; 4-6, 8, 19; 5-1, 7, 10, 22; 9-2
Order dependent options 4-17; 7-3
Order independent options 5-2, 12
Output file name 5-14

PACK command 3-5; 4-19
Pack working file 3-5; 4-15, 19
PACKNAM command 5-10
Packname specification 5-5, 14
Paper tape operations C-1
Paper tape punch control keys C-1
Paper tape reader 4-5
Paper tape reader control keys C-1
PARITY command 4-5
Parity switch 2-1; 6-4
PASSWOR command 4-20
Password 2-2; 4-16; 5-3; 6-5
Permanent file
    catalog 5-1
    command format 5-2
    commands 5-1, 6
    family (refer to family name)
    storage 3-3; 4-8
    mode of access 5-1, 2
    name 5-13
    types 5-1, 2

Permission 5-13, 19
Permission mode 5-3
PERMIT command 5-11
Post radix 4-2
PRIMARY command 4-20
Primary file 3-2, 3, 4, 6, 7; 4-6, 7, 9, 13, 14 15, 16, 19, 21; 5-7, 10, 12; 6-1
Private files 5-3, 18
Procedure file usage 4-22
Program editing 3-5; 6-1; 7-1, 2
Program execution 2-4; 3-6; 4-7, 8, 17
Project number (refer to CHARGE command)
PRU (physical record unit) 3-1
Public files 5-3, 21
Punch paper tape leader/trailer B-3; C-3
Punching a paper tape off-line C-2
Punching a paper tape on-line C-3
PURGE command 5-11

Question mark 6-3; 7-4

Read mode 5-3
Read with append mode 5-4
Read with modify mode 5-4
RECOVER command 4-20
Recovery 7-3, 5, 6, 7, 8, 9
Recovery of two or more users 7-7
Reformatting a file 8-2
Reformatting directives 8-3
Releasing permanent files (refer to PURGE command)
Releasing working files 3-2, 6, 7; 4-11
Remote job entry 4-21; 8-1
RENAME command 4-20
REPLACE command 4-8; 5-11; 7-2; 8-5
Requests for input 6-3; 7-4
RESEQ command 4-20
Resequencing line numbers 4-16; 7-2
Reserved file names F-1
RESOURC control card image 5-6, 15
RESTORE statement 3-3
RETURN command 4-21; 5-7
RETURN key 1-2; 6-2
REWIND command 3-3; 4-21
RFL control card image 9-1
RNH (run-no-header) command 4-23
ROUT command 4-5
RUN command 2-4; 3-6; 4-7, 21

SAVE command 3-3; 4-6, 8; 5-1, 10, 12; 9-2
Semiprivate file 5-3, 20
Set time limit 4-23
SETTL command 4-23; 7-3
S key 4-23
SORT command 3-4; 4-23
Sort flag 3-4; 4-19
Source code 3-3
SRU 2-7
Standard character set B-5, 6
Standard input lines 6-1
Standard print selection (refer to TERM command)
STATUS command (refer to ENQUIRE command)
Steps in job processing 6-4
STOP command 4-24; 7-4
SUBMIT command 4-24; 8-1
Submit error processing 8-5
Subsystem association with permanent files (refer to subsystem flag)
Subsystems available 1-2
Subsystem exit 9-2
Subsystem flag 3-3; 4-6, 8, 9; 5-12; 6-9
Subsystem selection, automatic 3-3, 4
Subsystem selection commands 2-2; 4-7
Successful recovery 7-6
Suspend job 4-12; 7-3
System commands 4-1
System malfunction 7-5
System resource units (SRU) 2-7

TAPE1 8-4
TAPE command 4-6; 7-3
Tape mode 4-5; C-3
TERM command 4-6
Terminals 1-1
Terminal
- control commands 4-2
- control keys B-3
- identification 2-2; 4-20; 6-4, 5
- job processing 6-4
- number 2-2; 4-20; 6-7; 7-6
- parity 4-5; 6-4
- print type (refer to TERM command)

Terminate
- executing program 6-3
- input line 6-2
- output 4-19
- program 4-19, 21; B-3

TEXT command 4-24
Text editing 3-5; 6-1; 7-1, 2
Text Editor 4-11; 6-1; 7-1
Text Editor commands E-1
Text mode 3-3, 4; 4-21
Time limit specification 4-19, 23; 7-4, 5
TIMEOUT 4-6
Time-sharing job commands 4-10

T, nnnnn 7-4
TRAN 4-24
Transparent mode 8-2
TTY (refer to TERM command)
Types of permanent files 5-1

Unsuccessful log-in 2-1; 6-6
Unsuccessful recovery 7-6
User number, entry of 2-2; 6-5

Validation
limits 4-13, 23
user 4-2, 13, 16, 17, 19, 23, 24, 25; 5-1, 5-5, 8, 11, 12, 15; 8-1

Working copy of file 5-1
Working files 3-2; 4-6, 9
Working file control 3-6
Write interlock 5-2
Write mode 5-3, 7, 8

X, nnnnn 9-2
X-OFF character C-2
X-ON character 4-5

COMMENT SHEET

MANUAL TITLE CDC KRONOS 2.1 Time-Sharing User's Reference Manual

PUBLICATION NO. 60407600 REVISION D

FROM: NAME:

BUSINESS ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

CUT ALONG LINE

AA3419 REV. 11/69 PRINTED IN U.S.A

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

CUT ALONG LINE

FOLD

FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINNESOTA 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION